

TUGAS AKHIR - KI141502

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Orang Pertama menggunakan Google Cardboard

KEN GENESIUS META BASUNDARA
NRP 5113100114

Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TUGAS AKHIR - KI141502

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Orang Pertama menggunakan Google Cardboard

KEN GENESIUS META BASUNDARA
NRP 5113100114

Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

(Halaman Ini Sengaja dikosongkan)



FINAL PROJECT- KI141502

Virtual Reality Game of Mosquito's Life from First Person View using Google Cardboard

KEN GENESIUS META BASUNDARA
NRP 5113100114

Advisor

Imam Kuswardayan, S.Kom., M.T.

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2017

(Halaman Ini Sengaja dikosongkan)

LEMBAR PENGESAHAN

PERMAINAN REALITAS VIRTUAL KEHIDUPAN HEWAN NYAMUK DARI SUDUT Pandang Orang Pertama Menggunakan Google Cardboard

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

KEN GENESIUS META BASUNDARA

NRP. 5113 100 114

Disetujui oleh Dosen Pembimbing Tugas Akhir
Imam Kuswardayan, S.Kom., M.T.
NIP: 19761215 200312 1 001

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom
NIP: 19771217 200312 2 001



**SURABAYA
MEI, 2017**

(Halaman Ini Sengaja dikosongkan)

PERMAINAN REALITAS VIRTUAL KEHIDUPAN HEWAN NYAMUK DARI SUDUT PANDANG ORANG PERTAMA MENGGUNAKAN GOOGLE CARDBOARD

Nama Mahasiswa : Ken Genesius Meta Basundara
NRP : 5113 100 114
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Imam Kuswardayan, S.Kom., M.T.
Dosen Pembimbing II : Dr. Eng Darlis Herumurti, S.Kom.,
M.Kom.

ABSTRAK

Permainan realitas virtual adalah permainan yang menggunakan sudut pandang orang pertama yang berkembang pesat akhir-akhir ini. Namun, saat ini permainan realitas virtual kebanyakan menjalankan simulasi kegiatan manusia dengan sudut pandang manusia. Belum ada permainan yang mensimulasikan kehidupan hewan dari sudut pandang hewan itu juga, khususnya di perangkat telepon genggam. Oleh karena itu, dibutuhkan permainan yang mensimulasikan kehidupan hewan dari sudut pandang hewan itu dengan harapan memperbanyak permainan yang sejenis serta mempelajari kehidupan hewan tersebut.

Permainan realitas virtual ini dapat menjadi solusi yang terbaik untuk mengatasi masalah tersebut. Dengan sudut pandang orang pertama atau sudut pandang hewan itu sendiri, pengguna bisa lebih merasakan dan mempelajari dengan menjadi hewan tersebut. Pada Tugas Akhir, penulis membangun permainan realitas virtual kehidupan hewan nyamuk dengan menggunakan Google Cardboard. Permainan ini mensimulasikan hewan nyamuk dalam hal mencari makan, yaitu dengan menghisap darah manusia. Permainan ini dijalankan di perangkat telepon pintar android dengan menggunakan Google Cardboard sebagai media realitas virtual dan controller jarak jauh sebagai media penggerak.

Uji coba permainan yang dibuat menggunakan metode kotak hitam dan kuesioner dari pengguna dengan mengujikan fungsionalitas permainan. Hasil kuesioner menunjukkan bahwa permainan yang dibuat lebih nyaman digunakan oleh pengguna yang sudah terbiasa menggunakan Google Cardboard dan terbiasa memainkan menggunakan controller. Dari hasil kuesioner juga didapatkan bahwa permainan sudah cukup nyaman, mudah digunakan, dan mendapatkan sensasi menjadi hewan yang dimainkan. Dengan dibangunnya permainan ini, diharapkan banyak industri pembuat permainan membuat game yang sejenis agar dapat dijadikan sebagai sarana pembelajaran kehidupan makhluk hidup lain dan memperbanyak permainan realitas virtual pada perangkat genggam.

Kata kunci: Realitas Virtual, Google Cardboard, Simulasi, Kehidupan Hewan

VIRTUAL REALITY GAME OF MOSQUITO'S LIFE FROM FIRST PERSON VIEW USING GOOGLE CARDBOARD

Student Name : Ken Genesius Meta Basundara
NRP : 5113 100 114
Major : Teknik Informatika FTIf-ITS
Advisor I : Imam Kuswardayan, S.Kom., M.T.
Advisor II : Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

ABSTRACT

Virtual reality games are games that use first-person view that is growing rapidly these days. However, today's virtual reality games mostly runs the simulation of human activity with a human point of view, there are no games that simulates animal's life from the animal's point of view, especially on mobile devices. Therefore, it needs a game that simulates animal's life from the animal's point of view in the hope increasing similar games and learning about the animal's life.

This virtual reality games can be the best solution for solving the problem. With first person view or animal view itself, user can feel or learn better by becoming the animal. In final project, writer build virtual reality games of mosquito's life using Google Cardboard. This game simulate mosquito in the case of feeding, which is by sucking human blood. This game run on Android smartphone with Google Cardboard as a virtual reality device and remote controller as control media.

The experiment of this game is based on black box method and questionnaire from users by testing game functionality. From the result of the questionnaire shows that the game is more convenient by user who used to play with Google Cardboard and remote controller. From the questionnaire result also found that game is quite comfortable, easy to use, and get the sensation of being animal that are played. With the making of this

game, it is expected many video games industry make similar games to be used for learning the lives of other living beings and increasing the virtual reality games on handheld devices.

Keywords: Virtual Reality, Google Cardboard, Simulation, Animal Life.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur kepada Allah Subhanahu wa ta'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Orang Pertama menggunakan Google Cardboard”.

Pengerjaan tugas akhir ini adalah momen bagi penulis untuk mengeluarkan seluruh kemampuan, hasrat, dan keinginan dari penulis sendiri dalam pemenuhan syarat untuk memperoleh gelar Sarjana Komputer untuk program studi Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.

Dalam pelaksanaan dan pembuatan tugas akhir ini penulis mendapatkan sangat banyak bantuan dari berbagai pihak. Melalui lembar ini, penulis ingin secara khusus menyampaikan ucapan terima kasih kepada:

1. Allah Subhanahu wa ta'ala yang telah melimpahkan rahmat, hidayah, dan inayah-Nya sehingga penulis menyelesaikan tugas akhir dengan baik.
2. Junjungan kepada Nabi Muhammad Shallahu Alayhi Wasallam yang telah menjadi inspirasi, panutan bagi seluruh pengikutnya termasuk penulis sehingga penulis tetap termotivasi dalam mengerjakan tugas akhir.
3. Keluarga penulis; bapak, ibu, dan kakak yang telah mencurahkan kasih sayang, perhatian, dan doa yang tiada hentinya kepada penulis.
4. Bapak Imam Kuswardayan, S.Kom., M.T. dan Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan bimbingan, motivasi, dan meluangkan waktu untuk membantu pengerjaan tugas akhir ini.
5. Bapak Dr. Eng. Darlis Herumurti, S.Kom, M.Kom. selaku ketua jurusan Teknik Informatika ITS, Bapak Radityo

Anggoro, S.Kom, M.Sc. selaku koordinator TA dan koordinator KP dan segenap Bapak/Ibu dosen Teknik Informatika yang telah memberikan ilmunya kepada penulis.

6. Seluruh staff Jurusan Teknik Informatika ITS yang telah mengayomi penulis selama berada di Teknik Informatika.
7. Teman-Teman IGS menemani dan membantu penulis saat mengerjakan tugas akhir.
8. Keluarga besar Teknik Informatika ITS angkatan 2013 yang telah menemani dan memberi pengalaman berharga bagi penulis sejak masuk Teknik Informatika sampai lulus.
9. Teman-teman angkatan 2011, dan 2012 yang sudah memberikan semangat, bantuan dan ilmu non akademik kepada penulis baik dalam kehidupan sehari-hari maupun pengerjaan tugas akhir ini.
10. Pihak-pihak lain yang belum disebutkan satu per satu yang juga turut membantu penulis menyelesaikan tugas akhir ini.

Penulis telah berusaha sebaik mungkin dalam mengerjakan tugas akhir ini, namun penulis memohon maaf terlebih dahulu apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan baik sengaja maupun tidak disengaja. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya. Akhir kata, penulis berharap tugas akhir ini bisa bermanfaat untuk para pembaca.

Surabaya, Mei 2017

Penulis

Ken Genesisus Meta Basundara

DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	4
1.7 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA.....	9
2.1 Virtual Reality (VR)	9
2.2 Google Cardboard	10
2.3 Unity 3D	11
2.4 Blender	12
2.5 Game Mister Mosquito.....	13
2.6 Pemodelan dan Animasi Tiga Dimensi (3D).....	15
2.7 First Person Shooter (FPS)	15
2.8 Microsoft Visual Studio	16
2.9 Hewan Nyamuk	16
BAB III ANALISIS DAN PERANCANGAN.....	19
3.1 Analisis Sistem	19
3.2 Perancangan Model 3D	20
3.3 Perancangan Perangkat Lunak.....	21
3.3.1 Deskripsi Umum Perangkat Lunak.....	21
3.3.2 Spesifikasi Kebutuhan Fungsional.....	22

3.3.3	Spesifikasi Kebutuhan Non-Fungsional	23
3.3.4	Karakteristik Pengguna.....	24
3.4	Perancangan Sistem.....	24
3.4.1	Perancangan Diagram Kasus Penggunaan.....	24
3.4.2	Perancangan Skenario Kasus Penggunaan	26
3.4.3	Perancangan Antarmuka Pengguna	38
3.4.4	Perancangan Alur Permainan.....	42
3.4.5	Perancangan Aturan Permainan.....	44
3.4.6	Perancangan Skenario Level Permainan.....	45
3.4.7	Perancangan Kontrol Permainan	47
BAB IV IMPLEMENTASI.....		49
4.1	Lingkungan Implementasi	49
4.2	Implementasi Lingkungan Ruang Virtual.....	50
4.2.1	Implementasi Ruang Virtual	51
4.2.2	Implementasi Objek Pendukung Ruang Virtual ..	55
4.3	Implementasi Objek Manusia.....	58
4.4	Implementasi Objek Pengguna.....	66
4.4.1	Implementasi Kamera Realitas Virtual.....	66
4.4.2	Implementasi Kontrol Pengguna	67
4.4.3	Implementasi <i>Collider</i> Pengguna.....	69
4.5	Implementasi Interaksi Objek Manusia.....	70
4.5.1	Implementasi <i>Collider</i> Objek Manusia.....	70
4.5.2	Implementasi Area Tempat Menghisap Darah	71
4.5.3	Implementasi Area Pandangan Objek Manusia.....	73
4.5.4	Implementasi Alur Animasi Objek Manusia	75
4.6	Implementasi Interaksi Objek Tak Bergerak.....	77
4.7	Implementasi Interaksi Objek Bergerak	79
4.8	Implementasi Alur Permainan	80
4.8.1	Implementasi Waktu Permainan	81
4.8.2	Implementasi Nyawa Objek Pengguna.....	81
4.8.3	Implementasi Melukai Objek Pengguna.....	82
4.8.4	Implementasi Kesadaran Objek Manusia	83
4.8.5	Implementasi Darah yang Terhisap dari Objek Manusia	84
4.8.6	Implementasi Racun	85

4.8.7	Implementasi <i>Minigame</i>	88
4.9	Implementasi Antarmuka	94
4.9.1	Implementasi Antarmuka Menu Utama.....	94
4.9.2	Implementasi Antarmuka Layar Tunggu	96
4.9.3	Implementasi Antarmuka Karakter Pengguna	100
4.9.4	Implementasi Antarmuka <i>Minigame</i>	109
BAB V PENGUJIAN DAN EVALUASI		111
5.1	Lingkungan Uji Coba	111
5.2	Pengujian Fungsionalitas.....	111
5.2.1	Skenario Pengujian Memilih Menu Navigasi	112
5.2.2	Skenario Pengujian Melihat Info Fakta (<i>Fun Fact</i>) dari Hewan Nyamuk	114
5.2.3	Skenario Pengujian Bergerak atau Terbang Mengelilingi Ruangan.....	115
5.2.4	Skenario Pengujian Menghisap Darah Musuh atau Target	117
5.2.5	Skenario Pengujian Menyelesaikan <i>Minigame</i>	119
5.2.6	Skenario Pengujian Menghindari Pukulan Musuh atau Target.....	121
5.2.7	Skenario Pengujian Bersembunyi dari Racun.....	123
5.3	Pengujian Aplikasi Terhadap Pengguna	125
5.3.1	Skenario Uji Coba oleh Pengguna	125
5.3.2	Daftar Penguji Permainan	126
5.3.3	Hasil Uji Coba Pengguna.....	126
5.4	Evaluasi	129
5.4.1	Evaluasi Pengujian Fungsionalitas	129
5.4.2	Evaluasi Pengujian Permainan.....	130
BAB VI KESIMPULAN DAN SARAN.....		133
6.1.	Kesimpulan.....	133
6.2.	Saran.....	133
DAFTAR PUSTAKA.....		135
LAMPIRAN HASIL KUESIONER.....		137
BIODATA PENULIS.....		147

(Halaman Ini Sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Cara Kerja Virtual Reality	9
Gambar 2.2 Model dan Penggunaan dari Google Cardboard.....	10
Gambar 2.3 Tampilan Antarmuka dari Unity3D.....	11
Gambar 2.4 Tampilan Aplikasi Blender.....	12
Gambar 2.5 Objek-Objek Dasar dalam Blender.....	13
Gambar 2.6 Tampilan Screenshot game Mister Mosquito	14
Gambar 2.7 Morfologi dari Hewan Nyamuk	17
Gambar 3.1 Diagram Kasus Penggunaan	25
Gambar 3.2 Diagram Aktivitas Memilih Menu Navigasi	32
Gambar 3.3 Diagram Aktivitas Melihat Info Fakta.....	33
Gambar 3.4 Diagram Aktivitas Bergerak atau Terbang Mengelilingi Ruangan	34
Gambar 3.5 Menghisap Darah Musuh atau Target	35
Gambar 3.6 Diagram Aktivitas Menyelesaikan <i>Minigame</i>	36
Gambar 3.7 Diagram Aktivitas Menghindari Pukulan Musuh atau Target	37
Gambar 3.8 Diagram Aktivitas Bersembunyi dari Racun.....	38
Gambar 3.9 Rancangan Antarmuka Menu Utama Permainan	39
Gambar 3.10 Rancangan Antarmuka Layar Tunggu (<i>Loading Screen</i>).....	40
Gambar 3.11 Rancangan Antarmuka Dalam Permainan (<i>Ingame</i>)	41
Gambar 3.12 Rancangan Antarmuka <i>Minigame</i>	42
Gambar 4.1 Tampilan “ <i>User Preferences</i> ” sebelum terinstall <i>add-on</i> Archimesh	50
Gambar 4.2 Tampilan “ <i>User Preferences</i> ” setelah terinstall <i>add- on</i> Archimesh	51
Gambar 4.3 Hasil <i>Extrude</i> dari Dinding dan Terbentuknya Kerangka Ruangan	52
Gambar 4.4 Hasil Bentuk Ruangan setelah Penambahan Atap dan Lantai.....	52
Gambar 4.5 Hasil Akhir Bentuk Ruangan.....	53

Gambar 4.6 Tampilan Ruangan Dalam Aplikasi Unity	54
Gambar 4.7 Tampilan Ruangan Setelah Diberi Warna dan Pencahayaan	55
Gambar 4.8 <i>Mirror Modifier</i> pada Objek Penyangga Tempat Tidur	56
Gambar 4.9 Hasil Akhir Objek Tempat Tidur.....	57
Gambar 4.10 Tampilan Tempat Tidur Dalam Aplikasi Unity.....	57
Gambar 4.11 Tampilan Rak Buku, Lampu Duduk dan Lemari Dalam Aplikasi Unity.....	58
Gambar 4.12 Tampilan Objek Manusia	59
Gambar 4.13 Hasil <i>UV Mapping</i> dari Objek Manusia	60
Gambar 4.14 Hasil Pewarnaan <i>UV Layout</i>	61
Gambar 4.15 Hasil Kerangka Tulang pada Objek Manusia.....	62
Gambar 4.16 Hasil Pembuatan Animasi pada Objek Manusia ...	63
Gambar 4.17 Penambahan Tekstur pada Objek Manusia	64
Gambar 4.18 Hasil Akhir Objek Manusia pada Aplikasi Unity..	64
Gambar 4.19 Hasil Akhir Objek Manusia pada Level 2	65
Gambar 4.20 Hasil Akhir Objek Manusia pada Level 3	65
Gambar 4.21 Tampilan Impor <i>Package</i> Integrasi Google Cardboard ke Unity	66
Gambar 4.22 Hasil Penambahan Kamera Realitas Virtual.....	67
Gambar 4.23 Hasil Penambahan Komponen <i>Character Controller</i>	68
Gambar 4.24 Hasil Penambahan <i>Sphere Collider</i>	70
Gambar 4.25 Hasil Penambahan <i>Collider</i> pada Seluruh Anggota tubuh Objek Manusia	71
Gambar 4.26 Hasil Penambahan Area Menghisap Darah pada Daerah Pipi Kiri Objek Manusia	73
Gambar 4.27 Hasil Penambahan Animator pada Objek Manusia	75
Gambar 4.28 Isi dari Objek Manusia termasuk Animasinya	76
Gambar 4.29 Hasil Alur Animasi pada Objek Manusia	77
Gambar 4.30 Hasil Penambahan <i>Collider</i> dan <i>Script</i> pada Objek Tak Bergerak	78
Gambar 4.31 Hasil Penambahan <i>Collider</i> dan <i>Script</i> pada Objek Bergerak	80

Gambar 4.32 Tampilan Antarmuka Menu Utama	94
Gambar 4.33 Tampilan Antarmuka Layar Tunggu	97
Gambar 4.34 Tampilan Antarmuka Karakter Pengguna	100
Gambar 4.35 Tampilan Antarmuka <i>Pause</i>	106
Gambar 4.36 Tampilan Antarmuka <i>Gameover</i>	107
Gambar 4.37 Tampilan Antarmuka <i>Level Complete</i>	108
Gambar 4.38 Tampilan Antarmuka <i>Minigame</i>	109
Gambar 5.1 Uji Coba saat Menekan Tombol Navigasi	113
Gambar 5.2 Uji Coba saat Selesai Menekan Tombol Navigasi	113
Gambar 5.3 Uji Coba saat Melihat Info Fakta Tentang Nyamuk	115
Gambar 5.4 Uji Coba saat Bergerak atau Terbang Mengelilingi Ruangan	116
Gambar 5.5 Uji Coba saat Selesai Bergerak atau Terbang Mengelilingi Ruangan	117
Gambar 5.6 Uji Coba saat Menghisap Darah Musuh atau Target	118
Gambar 5.7 Uji Coba saat Selesai Menghisap Darah Musuh atau Target	119
Gambar 5.8 Uji Coba saat Menyelesaikan <i>Minigame</i>	120
Gambar 5.9 Uji Coba saat Selesai Menyelesaikan <i>Minigame</i>	121
Gambar 5.10 Uji Coba saat Menghindari Pukulan Musuh atau Target	122
Gambar 5.11 Uji Coba saat Selesai Menghindari Pukulan Musuh atau Target	123
Gambar 5.12 Uji Coba saat Bersembunyi dari Racun	124
Gambar 5.13 Uji Coba saat Selesai Bersembunyi dari Racun	125
Gambar 8.1 Lembar Kuesioner Saddhana Arta Daniswara (1)	137
Gambar 8.2 Lembar Kuesioner Saddhana Arta Daniswara (2)	138
Gambar 8.3 Lembar Kuesioner Ishardan (1)	139
Gambar 8.4 Lembar Kuesioner Ishardan (2)	140
Gambar 8.5 Lembar Kuesioner Wildan Lutfi S.F.S (1)	141
Gambar 8.6 Lembar Kuesioner Wildan Lutfi S.F.S (2)	142
Gambar 8.7 Lembar Kuesioner Purbo Panambang (1)	143
Gambar 8.8 Lembar Kuesioner Purbo Panambang (2)	144

Gambar 8.9 Lembar Kuesioner Cayza Angka Maulana (1)	145
Gambar 8.10 Lembar Kuesioner Cayza Angka Maulana (2)	146

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna	24
Tabel 3.2 Skenario kasus penggunaan.....	25
Tabel 3.3 Skenario Kasus Penggunaan Memilih Menu Navigasi	27
Tabel 3.4 Skenario Kasus Penggunaan Melihat Info Fakta (<i>Fun Fact</i>) dari Hewan Nyamuk.....	27
Tabel 3.5 Skenario Kasus Penggunaan Bergerak atau Terbang Mengelilingi Ruangan	28
Tabel 3.6 Skenario Kasus Penggunaan Menghisap Darah Musuh atau Target.....	29
Tabel 3.7 Skenario Kasus Penggunaan Menyelesaikan <i>Minigame</i>	29
Tabel 3.8 Skenario Kasus Penggunaan Menghindari Pukulan Musuh atau Target.....	30
Tabel 3.9 Skenario Kasus Penggunaan Bersembunyi dari Racun	31
Tabel 3.10 Skenario Level Permainan.....	46
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak (1)	49
Tabel 4.2 Lingkungan Implementasi Perangkat Lunak (2)	49
Tabel 5.1 Lingkungan Perangkat Uji Coba	111
Tabel 5.2 Pengujian Memilih Menu Navigasi.....	112
Tabel 5.3 Pengujian Melihat Info Fakta (<i>Fun Fact</i>) dari Hewan Nyamuk	114
Tabel 5.4 Pengujian Bergerak atau Terbang Mengelilingi Ruangan.....	115
Tabel 5.5 Pengujian Menghisap Darah Musuh atau Target	117
Tabel 5.6 Pengujian Menyelesaikan <i>Minigame</i>	119
Tabel 5.7 Pengujian Menghindari Pukulan Musuh atau Target	121
Tabel 5.8 Pengujian Bersembunyi dari Racun	123
Tabel 5.9 Daftar Penguji Permainan	126
Tabel 5.10 Skala Penilaian Uji Coba.....	127

Tabel 5.11 Penilaian Kenyamanan Antarmuka dan Lingkungan Realitas Virtual.....	127
Tabel 5.12 Penilaian Kemudahan Permainan.....	128
Tabel 5.13 Penilaian Suasana Permainan.....	128
Tabel 5.14 Penilaian Performa Permainan	129
Tabel 5.15 Rekapitulasi Hasil Uji Coba Fungsionalitas.....	130
Tabel 5.16 Rangkuman Uji Coba Pengguna	130
Tabel 5.17 Rangkuman Uji Coba Pengguna	131

DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi Bergerak atau Terbang	69
Kode Sumber 4.2 Fungsi Interaksi Memasuki dan Keluar dari Area Menghisap Darah.....	72
Kode Sumber 4.3 Fungsi Interaksi Memasuki, Diam, dan Keluar dari Area Pandangan Objek Manusia.....	75
Kode Sumber 4.4 Fungsi Interaksi Memasuki, dan Keluar dari Area Sembunyi.....	78
Kode Sumber 4.5 Fungsi Penggerak Objek Bergerak.....	79
Kode Sumber 4.6 Fungsi Menjalankan dan me-reset Waktu Permainan.....	81
Kode Sumber 4.7 Fungsi Menjalankan dan Mengurangi Nyawa Pengguna	82
Kode Sumber 4.8 Fungsi Melukai Objek Pengguna	83
Kode Sumber 4.9 Fungsi Menjalankan dan Mengurangi Kesadaran Objek Manusia.....	84
Kode Sumber 4.10 Fungsi Menjalankan dan Menambah Darah.....	85
Kode Sumber 4.11 Fungsi pada <i>script Human.cs</i>	86
Kode Sumber 4.12 Fungsi pada <i>script Spray.cs</i>	87
Kode Sumber 4.13 Fungsi pada <i>script PoisonZone.cs</i>	87
Kode Sumber 4.14 Fungsi pada <i>script HealthManager.cs</i>	88
Kode Sumber 4.15 Fungsi pada <i>script RhythmMiniGame.cs</i>	90
Kode Sumber 4.16 Fungsi pada <i>script RhythmGameButton.cs</i> ...	92
Kode Sumber 4.17 Fungsi pada <i>script Hitzone.cs</i>	92
Kode Sumber 4.18 Fungsi pada <i>script Fly.cs</i>	94
Kode Sumber 4.19 Fungsi Tombol “New Game”	95
Kode Sumber 4.20 Fungsi Tombol “Select Level” dan Pemilihan Level.....	96
Kode Sumber 4.21 Fungsi Tombol “Exit”	96
Kode Sumber 4.22 Fungsi Menampilkan Fakta	97
Kode Sumber 4.23 Fungsi Menampilkan Perintah Bersiap-siap Memasuki Level.	99
Kode Sumber 4.24 Fungsi Proses Sistem Me-load Level	100
Kode Sumber 4.25 Fungsi Menampilkan Notifikasi	101

Kode Sumber 4.26 Fungsi Menampilkan Nyawa dan Ikon Racun	103
Kode Sumber 4.27 Fungsi Menampilkan Kesadaran dan Ikon Sembunyi.....	105
Kode Sumber 4.28 Fungsi Menampilkan Darah yang Dihisap.	105
Kode Sumber 4.29 <i>Script</i> Antarmuka <i>Pause</i>	106
Kode Sumber 4.30 <i>Script</i> Antarmuka <i>Gameover</i>	108
Kode Sumber 4.31 <i>Script</i> Antarmuka <i>Level Complete</i>	109
Kode Sumber 4.32 <i>Script</i> Menampilkan Arah pada <i>Minigame</i> .	110

BAB I

PENDAHULUAN

1.1 Latar Belakang

Video Game berbasis realitas virtual akhir-akhir ini berkembang pesat sejak adanya teknologi terbaru yang memaksimalkan sensasi bermain di dunia dalam game. Teknologi ini sudah diimplementasikan ke dalam game sejak pertengahan tahun 1990 yang diperkenalkan oleh perusahaan Nintendo. Pada saat itu teknologi yang digunakan masih berupa tampilan monochromatic dan sudut pandang yang sangat terbatas[1]. Seiring perkembangan teknologi, game berbasis realitas virtual mulai dikembangkan lagi dengan teknologi yang lebih canggih dan fitur yang lebih maksimal. Tidak hanya untuk home video game console, tetapi juga handheld console untuk merasakan sensasi realitas virtual dalam game dalam bentuk portabel.

Walaupun teknologi tentang realitas virtual yang semakin canggih, tidak banyak game yang dibuat dalam bentuk berbasis realitas virtual, khususnya di device portable seperti smartphone, dari website informasi vrgamesfor.com, dari 1134 game berbasis virtual reality yang dirilis, hanya sekitar 194 yang dirilis pada *smartphone*, itu juga tidak semua device realitas virtual bisa digunakan[2]. Meskipun ada, game yang dibuat kebanyakan mensimulasikan dalam keadaan tertentu, seperti simulasi menaiki *roller coaster*, terjebak di dalam suatu ruangan, berada di suasana perang, di ruang angkasa, dan lain-lain. Selain itu, game yang ada kebanyakan masih mensimulasikan aktifitas dari kehidupan manusia dan belum ada yang mensimulasikan bagaimana aktifitas dari kehidupan hewan. Padahal manusia sudah biasa menjalankan aktifitas yang bisa dilakukan di kehidupannya dari sudut pandang mereka sendiri, tetapi tidak bisa menjalankan aktifitas dari kehidupan hewan dengan cara pandang mereka.

Dari kekurangan tersebut, diperlukan untuk simulasi sekaligus permainan yang mensimulasikan aktifitas dari kehidupan hewan dari sudut pandang mereka untuk pembelajaran

dan sensasi permainan yang berbeda. Salah satu hewan yang bisa dipelajari yaitu nyamuk. Nyamuk adalah hewan yang dapat ditemukan dalam kehidupan sehari-hari manusia. Dapat dibilang keberadaan nyamuk cukup mengganggu manusia karena aktifitas mereka yaitu menghisap darah manusia. Tetapi tanpa disadari, aktifitas tersebut adalah cara makan mereka untuk bertahan hidup walaupun resiko kehilangan nyawa mereka sangat besar karena ukuran mereka yang sangat kecil sedangkan mangsa mereka sangat besar bagi pandangan mereka. Aktifitas tersebut dapat dijadikan sebuah permainan yang mensimulasikan cara bertahan hidup dari nyamuk. Permainan ini akan diimplementasikan dalam bentuk realitas virtual dengan menggunakan *head-up display* Google Cardboard untuk mendapatkan sensasi sudut pandang orang pertama dari hewan nyamuk.

Dari pembuatan game ini, diharapkan pengguna dapat mengerti dan merasakan bagaimana cara makan untuk bertahan hidup dari hewan nyamuk. Selain itu, pengguna juga mendapatkan sensasi permainan yang berbeda dari permainan yang ada dengan membuat pengguna merasakan bagaimana jadi hewan nyamuk diantara mangsa-mangsanya yang berukuran raksasa bagi hewan tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dikemukakan, beberapa permasalahan yang akan diselesaikan dalam tugas akhir adalah sebagai berikut:

1. Bagaimana rancangan aturan main, skenario, dan level dari permainan simulasi hewan nyamuk?
2. Bagaimana rancangan dari perilaku pada setiap objek yang terlibat pada game simulasi hewan nyamuk?

1.3 Batasan Masalah

Beberapa batasan masalah yang menjadi batas pada tugas akhir ini adalah sebagai berikut:

- Dikembangkan untuk smartphone bersistem operasi Android
- Penggunaan controller untuk penggerak permainan menggunakan Bluetooth controller gamepad
- Aplikasi dibangun dengan game engine Unity3D dengan bantuan Software Development Kit (SDK) Cardboard
- Objek 3D yang dibuat adalah ruangan kamar tidur, ruang keluarga dan properti-properti pendukung sesuai dengan ruangan tersebut, dan manusia sebagai target untuk menghisap darah dengan me-load file berekstensi fbx
- Menggunakan mode *first person shooter*.
- Manusia yang digunakan sebagai target untuk menghisap darah bergerak secara statis atau tidak mengikuti posisi nyamuk saat melakukan aktifitas
- Informasi trivia tentang fakta nyamuk hanya ditampilkan saat me-load level permainan
- Pengguna tidak dapat terbang keluar ruangan. Dengan kata lain hanya sampai tembok bagian dalam ruangan, sehingga bagian luar ruangan tidak akan ditampilkan

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini yaitu terciptanya game simulasi yang berbeda dengan cara mensimulasikan aktifitas mencari makan dari hewan nyamuk dari sudut pandang orang pertama atau hewan itu sendiri.

1.5 Manfaat

Manfaat dari tugas akhir ini antara lain:

1. Mendapatkan sensasi bermain yang berbeda dengan menggunakan sudut pandang hewan nyamuk.
2. Membuat pengguna merasakan bagaimana cara hewan nyamuk mencari makan dan menghindari mangsanya yang lebih besar dari nyamuk itu sendiri.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Pada tahap studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang akan digunakan. Literatur yang didapat berasal dari video dan artikel yang beredar di internet.

B. Perancangan perangkat lunak

Dalam melakukan perancangan pembuatan tugas akhir ini diperlukan 4 langkah:

1. *Modeling*

Pada tahap ini yaitu membuat objek yang dibutuhkan pada pembuatan tugas akhir. Objek ini bisa berbentuk objek dasar seperti *sphere* (bola), *cube* (kubus) sampai objek kompleks seperti manusia. Langkah pertama yang harus dilakukan untuk membuat objek adalah menentukan tipe objek dasar yang digunakan untuk membuat objek yang diinginkan. Pada pembuatan permainan realitas virtual ini diperlukan tipe objek dasar seperti kubus dan bola dalam pembuatan ruangan dan propertinya. Untuk tipe objek kompleks diperlukan beberapa gabungan objek dasar dan kerangka dalam pembuatan manusia.

2. *Animating*

Setelah tahap modeling selesai tahap kedua yang dilakukan adalah membuat animasi. Proses ini dilakukan pada objek-objek yang dapat berinteraksi dalam permainan realitas virtual. Objek-objek ini diberi animasi sesuai dengan fungsinya dan menghasilkan respon terhadap permainan

ini. Objek-objek yang dimaksud yaitu manusia, dan objek properti seperti obat nyamuk atau kipas angin.

3. *Programming*

Tahap ketiga yang dilakukan adalah membuat kode pemograman. Proses ini merupakan memasukkan objek-objek yang telah dibuat ke dalam *Game Engine* agar dapat berinteraksi dan memiliki fungsi dan peran terhadap permainan. Proses ini juga dilakukan untuk membuat interaksi kamera dalam permainan menjadi realitas virtual. Bahasa yang digunakan dalam proses ini adalah C#.

4. *Texturing*

Setelah tahap *programming* selesai tahap yang keempat yang dilakukan adalah proses *texturing*. Proses ini menentukan karakteristik dari objek menggunakan pola yang disebut *texture*. Objek-objek yang telah dibuat diberi warna serta diberi pola sehingga tampilan akan lebih realistis dan menarik.

C. Implementasi dan pembuatan sistem

Pada tahap ini dilakukan implementasi objek-objek dengan metode yang diusulkan dari rancangan yang telah dibuat pada tahap sebelumnya. Implementasi ini dilakukan dengan menggunakan aplikasi Unity dengan *library* Google Cardboard SDK dengan bahasa pemograman C# agar bisa berjalan di perangkat telepon genggam. Implementasi yang dibuat meliputi interaksi karakter utama dengan manusia, dan interaksi karakter utama dengan properti yang ada.

D. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk mencoba aplikasi dapat berjalan atau tidak. Pengujian yang dilakukan antara lain:

1. Pengujian *Blackbox*

Pengujian *blackbox* adalah pengujian yang berfokus pada spesifikasi fungsional dari Permainan yang telah dibuat, tester dapat mencoba kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional permainan. Pengujian ini dilakukan untuk menguji apakah objek 3D sudah sesuai dengan kebutuhan pengguna atau tidak.

2. Pengujian Usabilitas

Pengujian usabilitas dilakukan dengan cara melakukan survei ke tester untuk mengukur tingkat kepuasan dalam hal hiburan dari game yang dibuat.

- E. Penyusunan laporan tugas akhir
Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari permainan yang dibuat, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian

membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan permainan realitas virtual kehidupan nyamuk, dan antarmuka permainan.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh permainan dan evaluasi untuk mengetahui kemampuan permainan.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan permainan yang sejenis kedepannya.

(Halaman Ini Sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

2.1 Virtual Reality (VR)

Virtual Reality atau Realitas Virtual adalah teknologi komputer yang menggunakan perangkat lunak untuk menghasilkan gambar yang realistis, suara dan sensasi lain yang mereplika lingkungan nyata (atau membuat lingkungan dari imajinasi), dan mensimulasikan kehadiran fisik pengguna dalam lingkungan, dengan memungkinkan pengguna untuk berinteraksi dalam ruangan ini dan benda di dalamnya dengan menggunakan tampilan layar atau proyektor khusus dan perangkat lainnya [1]. VR dibagi dan secara obyektif hadir seperti dunia fisik, ditampilkan seperti sebuah karya seni, dan memiliki imajinasi yang tidak terbatas dan tidak berbahaya. Ketika VR tersebar di penjuru dunia, VR tidak akan dilihat sebagai media penengah yang digunakan dalam dunia nyata, melainkan sebagai dunia tambahan[3]. Tampilan konsep dan cara penggunaan dapat dilihat pada Gambar 2.1.



Gambar 2.1 Cara Kerja Virtual Reality

Dengan menggunakan Teknologi Realitas Virtual, pengguna dapat memanfaatkan inderanya untuk berinteraksi dengan lingkungan. Alat yang diperlukan untuk menampilkan

lingkungan realitas virtual disebut *Head-mounted display*, sedangkan alat yang dibutuhkan untuk menyesuaikan posisi dan orientasi pengguna yaitu *Tracker*.

2.2 Google Cardboard

Google Cardboard adalah alat virtual reality 3D yang dibuat dari karton yang diperkenalkan tahun 2015. Didesain oleh Google dan dibuat oleh pihak ketiga. Google Cardboard memegang smartphone Android dan menggunakan aplikasi Cardboard atau aplikasi pihak ketiga untuk menampilkan tampilan stereoscopic. Alat ini dikendalikan oleh gerakan kepala dan fitur accelerometer yang ada di smartphone[4]. Cardboard dijual dengan harga yang murah untuk merasakan sensasi realitas virtual. Bentuk Google Cardboard terlihat seperti pada Gambar 2.2.



Gambar 2.2 Model dan Penggunaan dari Google Cardboard

Untuk menggunakan Google Cardboard, dibutuhkan smartphone yang memiliki operasi sistem Android. Dan untuk pengembangan aplikasinya, Google menyediakan *Software Development Kit* sendiri (Google Cardboard SDK).

2.3 Unity 3D

Unity3D adalah mesin pengembangan game yang user-friendly dan powerful yang dapat dibuat di berbagai platform. Unity3D dapat digunakan dengan mudah untuk para pemula dan dapat digunakan secara maksimal untuk para ahli. Aplikasi ini dapat menarik perhatian siapa saja bagi yang ingin membuat game 3D untuk mobile, desktop, web, dan console[5]. Unity dibuat oleh Davor Helgason Nicholas pada tahun 2004. Aplikasi ini dibangun dengan fungsi standar umum yang digunakan dalam pembuatan game. Game engine ini juga menyediakan asset store untuk menambahkan asset dan fitur yang diinginkan pengguna. Tampilan antarmuka pada aplikasi Unity3D terlihat pada Gambar 2.3.



Gambar 2.3 Tampilan Antarmuka dari Unity3D

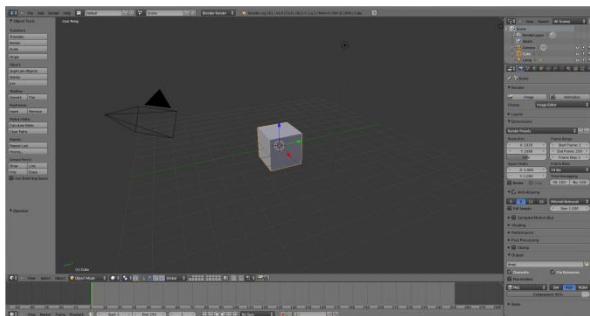
Unity 3D mendukung game berbentuk 3D dan 2D. Unity juga mendukung aplikasi pihak ketiga seperti Blender sebagai perangkat lunak desain dan animasi. Blender dapat digunakan untuk desain dan animasi objek yang dibutuhkan dalam game dikarenakan Unity tidak menyediakan fitur desain objek. Bahasa pemrograman yang digunakan untuk Unity adalah C# dan

Javascript. Pengguna dapat menggunakan bahasa pemrograman tersebut secara bersamaan dalam satu proyek atau memilih salah satu bahasa pemrograman tersebut.

2.4 Blender

Blender merupakan perangkat lunak untuk membuat pemodelan dan animasi tiga dimensi yang berbasis bebas bayar. Karena bersifat gratis, perangkat lunak ini bebas digunakan oleh siapa saja tanpa perlu membayar lisensi. Blender menggunakan bahasa pemrograman C, C++ dan python [6]. Bahasa utama yang digunakan adalah bahasa pemrograman python.

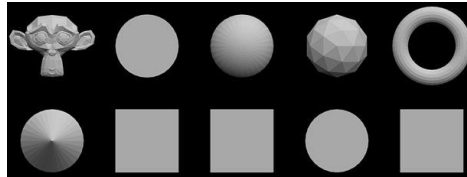
Tampilan UI dari Blender ini cukup banyak fungsi tombol, sehingga diperlukan pembelajaran dari awal baik orang yang pertama kali menggunakan maupun yang sudah terbiasa membuat model tiga dimensi. Namun, apabila setelah terbiasa dan mengetahui semua fungsi tombol yang ada maka Blender akan menjadi perangkat lunak yang mudah digunakan. Tampilan antarmuka dari aplikasi ini ditampilkan pada Gambar 2.4.



Gambar 2.4 Tampilan Aplikasi Blender

Seperti pada perangkat lunak tiga dimensi lainnya, pada Blender pembentukan objek dibuat dari objek-objek dasar. Yang termasuk objek-objek dasar adalah kubus, bidang datar, kerucut, lingkaran, dan tabung. Dengan memanipulasi objek dasar tersebut

dapat dibuat objek tiga dimensi yang diinginkan. Untuk mempermudah dalam memanipulasi objek, aplikasi memiliki fitur untuk melihat tampilan dari berbagai sudut pandang. Dengan begitu, dapat terlihat jelas semua bagian dari objek yang akan dibuat. Tampilan objek dasar dalam aplikasi Blender ditampilkan pada Gambar 2.5.



Gambar 2.5 Objek-Objek Dasar dalam Blender

Dalam Blender terdapat istilah *vertices*, *edge*, *face*, dan *tris*. *Vertice* merupakan objek berupa titik. *Edge* merupakan garis rusuk dari suatu objek. *Face* adalah sisi dari suatu objek yang terbentuk minimal dari tiga *vertice* yang saling terhubung, dan *tris* adalah bidang segitiga yang membentuk sisi atau *face* dari suatu objek. Hal yang dapat dimanipulasi dalam membuat objek tiga dimensi yaitu *vertices*, *edge*, dan *face*. Sedangkan *tris* digunakan untuk mengukur seberapa kompleks objek itu dibuat.

2.5 Game Mister Mosquito

Mister Mosquito adalah video game yang dikembangkan oleh ZOOM Inc untuk video game console PlayStation 2 (PS2). Permainan ini dirilis di Jepang oleh Sony pada 21 Juni 2001 dan Maret di wilayah lainnya sebagai bagian dari label Eidos Interactive “Fresh Games”[7]. Aturan bermain dari game ini adalah pemain mengontrol nyamuk bernama “Mister Mosquito” yang telah tinggal di rumah keluarga Yamada, Manusia yang akan menjadi sumber makanan dari pemain. Tujuan game ini adalah mengisi persediaan darah untuk bertahan hidup di musim dingin nanti. Pemain ditugaskan menghisap darah bagian tubuh

tertentu dari anggota keluarga tanpa diketahui. Jika pemain tidak berhati-hati, manusia akan menjadi stress dan akhirnya menyerang pemain[7]. Game ini digunakan sebagai referensi untuk game simulasi hewan nyamuk yang dibuat pada tugas akhir ini. Aturan main dari game ini akan dimodifikasi sesuai dengan kebutuhan game simulasi untuk tugas akhir. Pada tugas akhir yang dibuat, akan menggunakan sudut pandang orang pertama untuk bermain dengan menjadi nyamuk itu. Selain itu, manusia akan bergerak secara statis sesuai dengan animasi yang dibuat, sehingga tidak mengikuti posisi pemain. Pada game referensi ini memiliki *minigame* berupa menggerakkan stik analog untuk menghisap darah manusia, sedangkan pada tugas akhir yang dibuat, *minigame* berupa tombol arah yang harus ditekan sesuai dengan perintah. Mister Mosquito ternominasi game teraneh yang pernah ada oleh GameInformer, dan juga dinominasikan game yang lebih murah daripada terapi oleh GamesRadar sebagai obat untuk ketakutan terhadap serangga (Entomophobia)[8]. Tampilan *gameplay* permainan Mister Mosquito bisa dilihat pada gambar 2.6.



Gambar 2.6 Tampilan Screenshot game Mister Mosquito

2.6 Pemodelan dan Animasi Tiga Dimensi (3D)

Tiga dimensi atau bisa disingkat 3D atau yang biasa disebut ruang, adalah bentuk yang memiliki panjang, lebar, dan tinggi. Sehingga bentuk tersebut dapat dilihat dari segala sisi. Istilah biasanya digunakan dalam bidang seni, animasi, komputer, dan matematika [9].

Pemodelan adalah membentuk suatu benda-benda atau objek. Membuat dan mendesain objek tersebut sehingga terlihat seperti hidup. Sesuai dengan objek dan basisnya, proses ini secara keseluruhan dikerjakan di komputer. Melalui konsep dan proses desain, keseluruhan objek bisa diperlihatkan secara tiga dimensi, sehingga banyak yang menyebut hasil ini sebagai pemodelan tiga dimensi.

Animasi adalah membuat gerakan atau interaksi dari objek yang telah dibuat. Dengan animasi, objek akan bergerak seperti makhluk hidup dan dapat berinteraksi dengan objek lain. Animasi dibuat dengan menyimpan posisi, rotasi, dan skala dari kerangka dan model objek tiga dimensi dan menyimpannya ke dalam *frame*. Setelah beberapa *frame* berisi data dari objek tersebut, animasi pun terbuat. Animasi biasanya dibuat dengan kecepatan animasi 30 atau 60 *frame per second*.

2.7 First Person Shooter (FPS)

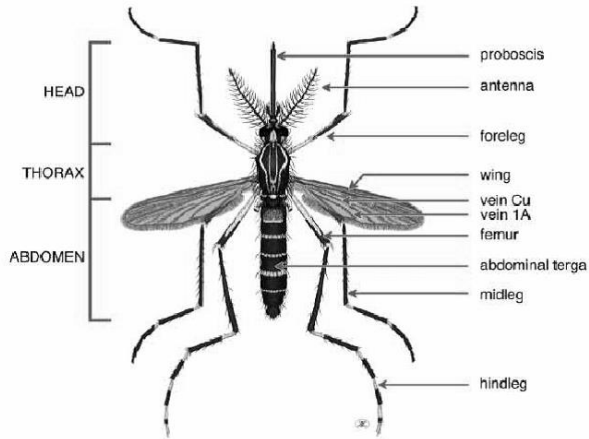
FPS merupakan genre action dalam video game, dimana sudut pandangnya berada pada sisi protagonis. FPS merupakan tipe game yang menampilkan apa yang orang sebenarnya lihat dan lakukan pada sebuah game. Pada umumnya FPS, menampilkan pandangan mata pemain kedepan dan hanya memperlihatkan tangan karakter yang membawa senjata dan perlengkapan di bawah layar. Permainan ini menggunakan pergerakan kedepan, kebelakang dan kesamping dengan menggunakan game controller. Untuk merubah sudut pandangan dapat dilakukan dengan menggerakkan mouse atau game controller[10].

2.8 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari compiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi yang biasa digunakan untuk mengembangkan aplikasi personal, aplikasi web, aplikasi perangkat bergerak, aplikasi console maupun komponen aplikasi pada Microsoft Visual Studio. Banyak compiler yang disediakan oleh aplikasi ini, misalnya saja Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Banyak pengembang aplikasi yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi yang disediakan, sehingga pengembang lebih mudah dan sangat terbantu dalam mengembangkan maupun membuat aplikasi mereka[11].

2.9 Hewan Nyamuk

Nyamuk adalah serangga tergolong dalam order Diptera, genera yang paling umum yaitu Anopheles, Culex, Psorophora, Ochlerotatus, Aedes, Sabethes, Wyeomyia, Culiseta, dan Haemagogus dari jumlah keseluruhan sekitar 35 genera yang merangkum 2700 spesies. Nyamuk mempunyai dua sayap bersisik, tubuh yang langsing, dan enam kaki panjang, antarspesies berbeda-beda tetapi jarang sekali melebihi 15 mm. Dalam bahasa Inggris, nyamuk dikenal sebagai "Mosquito", berasal dari sebuah kata dalam bahasa Spanyol atau bahasa Portugis yang berarti lalat kecil. Penggunaan kata Mosquito bermula sejak tahun 1583. Di Britania Raya nyamuk dikenal sebagai gnats[12]. Morfologi dari hewan nyamuk dapat dilihat pada Gambar 2.7.



Gambar 2.7 Morfologi dari Hewan Nyamuk

Nyamuk bertahan hidup dengan mencari makan dengan cara menghisap nektar dan menghisap darah. Nyamuk yang menghisap darah adalah nyamuk betina. Nyamuk betina inilah yang dijadikan topik utama dalam tugas akhir ini. Nyamuk betina menggunakan bagian mulutnya membentuk *proboscis* panjang untuk menembus kulit mangsanya untuk menghisap darah. Nyamuk betina membutuhkan darah untuk memenuhi kebutuhan pangannya dan protein untuk pembentukan telur. Darah yang terhisap akan disimpan didalam abdomen dan diproses untuk menghasilkan sel telur. Nyamuk paling tertarik dengan manusia yang memiliki golongan darah O, yang memiliki temperatur tubuh tinggi, dan ibu hamil.

(Halaman Ini Sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas tentang analisis dan perancangan permainan realitas virtual kehidupan hewan nyamuk dengan menggunakan *google cardboard*. Permainan ini bergenre *First Person Shooter (FPS)* dimana pemain melihat dari sudut pandang orang pertama dengan teknologi *Virtual Reality*. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Sistem

Permainan akan dibuat dengan teknologi *virtual reality*, yang artinya permainan akan dibuat dalam bentuk tiga dimensi. Permainan ini diimplementasikan pada perangkat telepon genggam dengan sistem operasi Android dikarenakan portabilitasnya yang tinggi, sehingga dapat digunakan dimana saja. Permainan ini diperlukan perangkat mobile yang memiliki sensor *gyroscope*, sensor magnet, *Graphic Processing Unit* yang mendukung *OpenGL E.S 2.0*, dan *Bluetooth* versi 2.0 yang seharusnya sudah dimiliki perangkat mobile saat ini.

Permainan realitas virtual cukup diminati oleh masyarakat karena dapat memberikan suasana dan sensasi yang lebih nyata dan lebih menarik. Permainan ini akan menggunakan alat *google cardboard* sebagai media realitas virtualnya, dimana alat ini menggunakan ponsel pintar sebagai media penampil realitas virtual, sedangkan penerima input digunakan sensor *gyroscope* untuk mendapatkan posisi dan rotasi pandangan pemain dalam permainan.

Untuk masukan perintah dalam permainan ini didapatkan dengan menggunakan *Bluetooth controller*, dimana controller ini disambungkan ke ponsel pintar menggunakan jaringan nirkabel *Bluetooth*. Controller ini digunakan sebagai alat untuk menggerakkan pemain ke posisi yang diinginkan pemain. Alat ini

juga digunakan untuk mengakses menu dan memberi perintah ke dalam permainan.

Permainan ini dibuat dengan menggunakan aplikasi Unity3D *game engine*. Aplikasi ini dapat membuat permainan tiga dimensi dari awal. Aplikasi ini juga mendukung beberapa platform dan teknologi realitas virtual yang dibutuhkan dalam permainan yang dibuat. Untuk membuat permainan realitas virtual, diperlukan tambahan *Software Development Kit* untuk *google cardboard*. Unity3D ini menggunakan bahasa pemrograman C#, yang nantinya akan digunakan untuk memberi fungsi dan tugas pada objek-objek yang telah dibuat.

3.2 Perancangan Model 3D

Objek-objek permainan ini dibangun dengan menggunakan aplikasi Blender versi 2.78. Objek-objek yang dibuat berupa objek bergerak dan objek mati. Objek memiliki ekstensi file fbx yang berisikan model, warna material yang digunakan, dan *texture* dari objek tersebut. Setiap objek memiliki fungsi masing-masing sehingga permainan dapat menjadi lebih nyata dan lebih banyak interaksi yang dilakukan.

Objek yang bergerak adalah objek berbentuk manusia, dan properti yang bergerak sendiri atau digerakkan oleh objek manusia. Objek manusia ini memiliki fungsi sebagai mangsa untuk dihisap darahnya dalam permainan. Objek ini memiliki *texture* setiap anggota badannya untuk dipoles agar lebih hidup. Objek ini juga memiliki animasi gerakan sendiri, seperti bermain komputer, melihat televisi, tidur, atau memukul pemain. Animasi ini akan memberi efek interaksi terhadap pemain dalam permainan sehingga permainan lebih interaktif dan menantang. Untuk objek properti adalah objek yang bergerak sendiri atau digerakkan oleh objek manusia, seperti obat nyamuk, kipas angin, dan alat pembunuh nyamuk. Alat ini dapat bergerak sendiri atau digerakkan oleh manusia dengan menggunakan animasi. Alat ini juga berfungsi untuk memberi efek interaksi terhadap pemain dalam permainan.

Objek yang tidak bergerak atau mati adalah objek properti. Objek ini tidak memiliki fungsi spesifik, hanya sebagai penghias ruangan atau tempat bersembunyinya pemain. Objek properti ini memiliki berbagai model seperti sofa, televisi, tempat tidur, meja, lemari, dan lain-lain. Objek ini hanya membuat ruangan menjadi lebih hidup dan berisi.

Objek-objek yang dibuat memiliki pola yang disebut dengan *texture*. Pola ini akan dipoles dengan menggunakan aplikasi Adobe Photoshop CS6 versi *trial*. Pola ini memberikan tekstur terhadap objek sehingga terlihat lebih hidup. *Texture* ini berekstensi *png*, dan dimasukkan ke kerangka objek yang disebut *UV Map* yang sudah tersimpan dalam ekstensi objek tersebut. Setelah *texture* dari objek selesai dibuat, objek dan *texture* akan dimasukkan ke dalam Unity3D *game engine* untuk disatukan dan dimasukkan dalam permainan.

3.3 Perancangan Perangkat Lunak

Pada subbab ini akan dibahas mengenai deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional dan kebutuhan non-fungsional serta bagaimana karakteristik pengguna permainan.

3.3.1 Deskripsi Umum Perangkat Lunak

Tugas akhir yang akan dibuat adalah permainan realitas virtual kehidupan hewan nyamuk dari sudut pandang orang pertama dengan menggunakan *google cardboard*. Permainan ini dikembangkan menggunakan *google cardboard SDK* sebagai *library* untuk bisa menjalankan teknologi realitas virtual. Permainan ini direncanakan berjalan pada perangkat ponsel pintar dengan sistem operasi Android. *Library* ini membutuhkan perangkat ponsel pintar yang sudah mendukung fitur gyroscope untuk mendeteksi posisi dan rotasi pandangan pemain, dan Bluetooth untuk mendapatkan input dari controller nirkabel dalam permainan realitas virtual yang dibuat.

Permainan ini akan menggunakan kamera sudut pandang orang pertama sebagai hewan nyamuk yang berada di suatu ruangan bersama target mereka yaitu manusia. Ruangan yang divisualisasikan dalam permainan ini seperti ruangan kamar tidur, ruangan keluarga, dan ruang makan. Bentuk dari ruangan, manusia beserta objek pendukung di dalamnya divisualisasikan dalam bentuk tiga dimensi.

Permainan ini dimainkan dengan menggerakkan karakter utama dengan cara terbang menggunakan controller untuk terbang maju, mundur, keatas, atau ke bawah. Permainan ini memiliki tujuan untuk menghisap darah dari target yaitu manusia. Pemain harus mencari posisi tempat tertentu untuk menghisap darah. Setelah pemain berhasil menemukan dan mendekati tempat menghisap darah tersebut, *minigame* akan berjalan. *Minigame* berupa menekan tombol sesuai dengan apa yang muncul pada layar. Setelah *minigame* terselesaikan dan tidak ada tempat mengisap darah lagi, game akan selesai dan pemain dapat melanjutkan ke level selanjutnya.

Rintangan dari permainan ini ialah target atau manusia itu sendiri, waktu level dan objek bergerak yang dapat mengganggu pemain. Manusia tersebut bisa memukul pemain apabila manusia merasa terganggu karena karakter utama berada di dekatnya. Manusia juga akan menyempatkan obat nyamuk dimana itu akan menjadi racun dan mengurangi darah karakter utama. Permainan juga diberi waktu bermain pada level yang sedang dimainkan. Apabila pemain terlalu lama tidak mencapai target, maka permainan selesai. Selain itu, pemain juga harus menghindari objek yang bergerak seperti kipas angin yang dapat mengganggu terbangnya karakter utama.

3.3.2 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem yang disampaikan sebelumnya, maka kebutuhan fungsional dalam permainan ini adalah pengguna dapat menggerakkan karakter utama dengan

cara terbang, pengguna dapat menghisap darah target yaitu manusia, serta dapat memainkan *minigame* saat menghisap darah.

3.3.3 Spesifikasi Kebutuhan Non-Fungsional

Pada permainan ini terdapat beberapa kebutuhan non-fungsional yang bertujuan meningkatkan kualitas dari permainan ini. Berikut daftar kebutuhan non-fungsional:

1. Grafis

Grafis merupakan aspek penting dalam kualitas permainan realitas virtual ini. Permainan harus menampilkan grafis senyaman mungkin dan senyata mungkin, sehingga pemain dapat merasakan sensasi menjadi karakter utama itu sendiri dan berada di lingkungan seolah berada di dunia nyata. Permainan ini menggunakan objek tiga dimensi, sehingga objek yang dibuat harus dibuat senyata mungkin dan nyaman untuk dilihat.

2. Performa

Permainan harus dapat berjalan mulus tanpa adanya hambatan atau *lag*. Permainan juga harus *me-load* *me-render* secepat mungkin untuk bisa dimainkan. Untuk meningkatkan performa, perlu mengoptimalkan jumlah objek tiga dimensi yang dibuat serta penggunaan cahaya dan bayangan dalam permainan yang optimal.

3. Kenyamanan dalam penggunaan controller

Permainan harus dapat membuat pemain nyaman dalam menggunakan controller. Controller yang digunakan pada permainan ini adalah controller nirkabel menggunakan *bluetooth*. Controller harus dapat menggerakkan karakter utama dengan bebas dan nyaman serta dapat memilih menu dengan tepat saat pemain masih menggunakan *google cardboard* dan memegang controller sebagai kendalinya, sehingga pemain tidak kebingungan saat memilih menu dan memegang controller ketika pemain tidak melihat dunia nyata.

3.3.4 Karakteristik Pengguna

Berdasarkan deskripsi umum diatas, maka diketahui bahwa pengguna yang akan menggunakan permainan ini hanya satu orang, yaitu pengguna yang menjalankan permainani. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

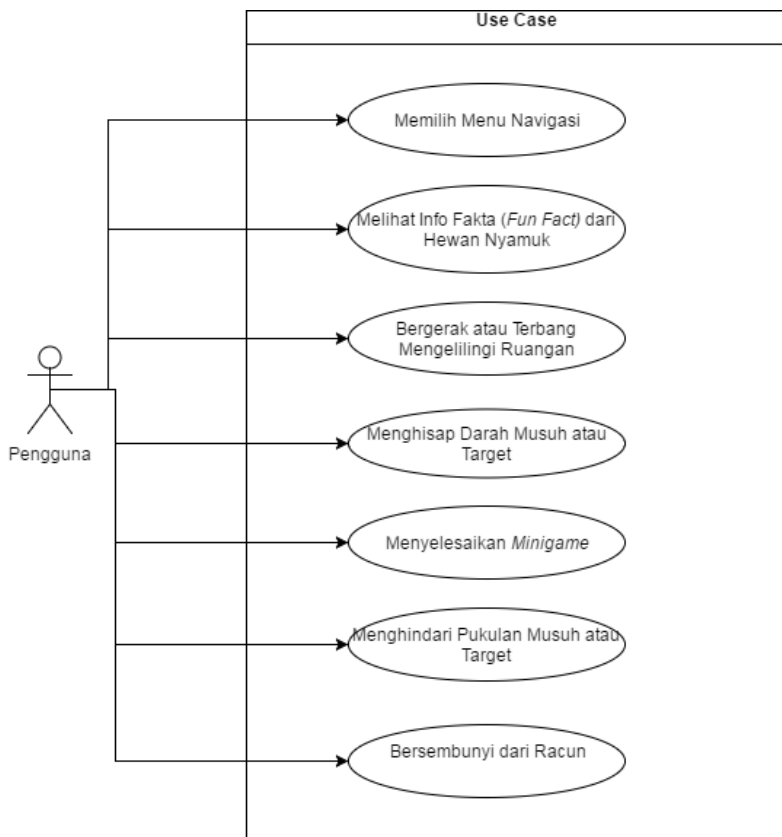
Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang menggunakan permainan	Menjalankan permainan	Pernah menggunakan <i>google cardboard</i> sebelumnya

3.4 Perancangan Sistem

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan antarmuka pengguna, dan perancangan kontrol permainan.

3.4.1 Perancangan Diagram Kasus Penggunaan

Dalam permainan yang dibuat pada tugas akhir ini, terdapat tujuh kasus penggunaan antara lain memilih menu navigasi, melihat info fakta (*fun fact*) dari hewan nyamuk, bergerak atau terbang mengelilingi ruangan, menghisap darah musuh atau target, menyelesaikan *minigame*, menghindari pukulan musuh atau target, dan bersembunyi dari racun. Diagram kasus penggunaan dari penjabaran diatas dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Kasus Penggunaan

Penjelasan singkat dari masing-masing kasus penggunaan dapat dilihat pada Tabel 3.2

Tabel 3.2 Skenario kasus penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Memilih menu navigasi	Pengguna dapat memilih menu navigasi saat permainan dijalankan

			pertama kali
2	UC-002	Melihat info fakta (<i>fun fact</i>) dari hewan nyamuk	Pengguna dapat melihat informasi tentang fakta dari hewan nyamuk saat permainan me-load level permainan
3	UC-003	Bergerak atau terbang mengelilingi ruangan	Pengguna dapat bergerak dengan bebas mengelilingi ruangan virtual dalam permainan
4	UC-004	Menghisap darah musuh atau target	Pengguna dapat menghisap darah target ketika pengguna berada di daerah tempat hisap darah dari target
5	UC-005	Menyelesaikan <i>minigame</i>	Pengguna dapat menjalankan dan menyelesaikan <i>minigame</i> ketika pengguna memulai menghisap darah dari target
6	UC-006	Menghindari pukulan musuh atau target	Pengguna dapat menghindari serangan atau pukulan dari target ketika target merasa terganggu dengan keberadaan pengguna
7	UC-007	Bersembunyi dari racun	Pengguna dapat bersembunyi di tempat yang aman ketika target menggunakan obat nyamuk untuk meyebarakan racun

3.4.2 Perancangan Skenario Kasus Penggunaan

Setelah diketahui bagaimana perancangan kasus penggunaan, langkah selanjutnya adalah membuat skenario tiap kasus penggunaan yang dibuat beserta diagram aktifitasnya. Skenario dan diagram tiap kasus penggunaan dapat dilihat pada subbab berikut.

3.4.2.1 Skenario Kasus Penggunaan

Skenario kasus penggunaan dibagi menjadi tujuh bagian sesuai dengan kasus penggunaan yang telah dibuat dan dapat dilihat pada subbab berikut ini.

3.4.2.1.1 Skenario Memilih Menu Navigasi

Skenario kasus penggunaan memilih menu navigasi dapat dilihat pada tabel 3.3.

Tabel 3.3 Skenario Kasus Penggunaan Memilih Menu Navigasi

Nama Kasus Penggunaan	Memilih Menu Navigasi
Kode	UC-001
Deskripsi	Pengguna dapat memilih menu navigasi saat permainan dijalankan pertama kali
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke permainan
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol yang ada dalam permainan 2. Sistem mengakses menu selanjutnya dan memasuki level
Alur Kejadian Alternatif	-

3.4.2.1.2 Skenario Melihat Info Fakta (*Fun Fact*) dari Hewan Nyamuk

Skenario kasus penggunaan melihat info fakta dari hewan nyamuk dapat dilihat pada tabel 3.4.

Tabel 3.4 Skenario Kasus Penggunaan Melihat Info Fakta (*Fun Fact*) dari Hewan Nyamuk

Nama Kasus Penggunaan	Melihat Info Fakta (<i>Fun Fact</i>) dari Hewan Nyamuk
------------------------------	--

Kode	UC-002
Deskripsi	Pengguna dapat melihat informasi tentang fakta dari hewan nyamuk saat permainan <i>me-load</i> level permainan
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu permainan baru atau memilih level 2. Sistem <i>me-load</i> object pada level tersebut dan menampilkan fakta dari hewan nyamuk. 3. Pengguna melihat fakta dari hewan nyamuk
Alur Alternatif	-

3.4.2.1.3 Skenario Bergerak atau Terbang Mengelilingi Ruangan

Skenario kasus penggunaan bergerak atau terbang mengelilingi ruangan dapat dilihat pada tabel 3.5.

Tabel 3.5 Skenario Kasus Penggunaan Bergerak atau Terbang Mengelilingi Ruangan

Nama Kasus Penggunaan	Bergerak atau Terbang Mengelilingi Ruangan
Kode	UC-003
Deskripsi	Pengguna dapat bergerak dengan bebas mengelilingi ruangan dalam permainan
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi dan objek sudah <i>di-load</i> dan <i>di-render</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menekan analog ke arah yang diinginkan untuk mengelilingi ruangan 2. Sistem menggerakkan karakter utama untuk terbang mengelilingi ruangan sesuai arah yang diinginkan pengguna
Alur Alternatif	-

3.4.2.1.4 Skenario Menghisap Darah Musuh atau Target

Skenario kasus penggunaan menghisap darah musuh atau target dapat dilihat pada tabel 3.6.

Tabel 3.6 Skenario Kasus Penggunaan Menghisap Darah Musuh atau Target

Nama Kasus Penggunaan	Menghisap Darah Musuh atau Target
Kode	UC-004
Deskripsi	Pengguna dapat menghisap darah target ketika pengguna berada di daerah tempat hisap darah dari target
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi dan objek sudah di-load dan di-render
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna mengarahkan karakter utama ke tempat area menghisap darah musuh 2. Pengguna menekan tombol untuk memulai menghisap darah musuh 3. Sistem memulai kejadian menghisap darah musuh
Alur Alternatif	-

3.4.2.1.5 Skenario Menyelesaikan *Minigame*

Skenario kasus penggunaan menyelesaikan *minigame* dapat dilihat pada tabel 3.7.

Tabel 3.7 Skenario Kasus Penggunaan Menyelesaikan *Minigame*

Nama Kasus Penggunaan	Meneylesaikan <i>Minigame</i>
Kode	UC-005

Deskripsi	Pengguna dapat menjalankan dan menyelesaikan minigame ketika pengguna memulai menghisap darah dari target
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi dan objek sudah di-load dan di-render
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna mengarahkan karakter utama ke tempat area menghisap darah musuh 2. Pengguna menekan tombol untuk memulai menghisap darah musuh 3. Sistem memulai <i>minigame</i> 4. Pengguna memainkan dan menyelesaikan <i>minigame</i> 5. Sistem menampilkan level selesai
Alur Alternatif	-

3.4.2.1.6 Skenario Menghindari Pukulan Musuh atau Target

Skenario kasus penggunaan menghindari pukulan musuh atau target dapat dilihat pada tabel 3.8.

Tabel 3.8 Skenario Kasus Penggunaan Menghindari Pukulan Musuh atau Target

Nama Kasus Penggunaan	Menghindari Pukulan Musuh atau Target
Kode	UC-006
Deskripsi	Pengguna dapat menghindari serangan atau pukulan dari target ketika target merasa terganggu dengan keberadaan pengguna
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi dan objek sudah di-load dan di-render
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna mengarahkan karakter utama ke tempat pandangan musuh 2. Sistem membuat musuh menyerang pengguna 3. Pengguna menghindar dari serangan musuh <p>A1. Pengguna gagal menghindar dari serangan</p>

	musuh
Alur Alternatif	A1. Pengguna gagal menghindari dari serangan musuh 1. Sistem mengurangi nyawa pengguna 2. Kembali ke alur kejadian nomor 1

3.4.2.1.7 Skenario Bersembunyi dari Racun

Skenario kasus penggunaan bersembunyi dari racun dapat dilihat pada tabel 3.9.

Tabel 3.9 Skenario Kasus Penggunaan Bersembunyi dari Racun

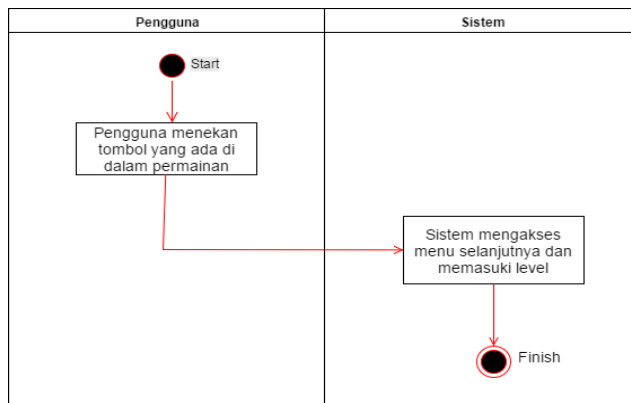
Nama Kasus Penggunaan	Bersembunyi dari Racun
Kode	UC-007
Deskripsi	Pengguna dapat bersembunyi di tempat yang aman ketika target menggunakan obat nyamuk untuk meyebarakan racun
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi dan objek sudah di-load dan di-render
Alur Normal	1. Pengguna mengarahkan karakter utama ke tempat pandangan musuh 2. Sistem membuat musuh menyerang pengguna 3. Pengguna membuat musuh terganggu 4. Sistem membuat musuh menyembrotkan obat nyamuk dan racun menyebar 5. Pengguna menghindari dan bersembunyi dari racun A1. Pengguna gagal menghindari dan bersembunyi dari racun
Alur Alternatif	A2. Pengguna gagal menghindari dan bersembunyi dari racun 3. Sistem mengurangi nyawa pengguna 4. Kembali ke alur kejadian nomor 1

3.4.2.2 Diagram Aktivitas

Diagram aktivitas menampilkan langkah-langkah normal yang harus dilakukan pemain untuk menjalankan kasus penggunaan aplikasi. Diagram aktifitas dapat dilihat pada subbab berikut.

3.4.2.2.1 Diagram Aktifitas Memilih Menu Navigasi

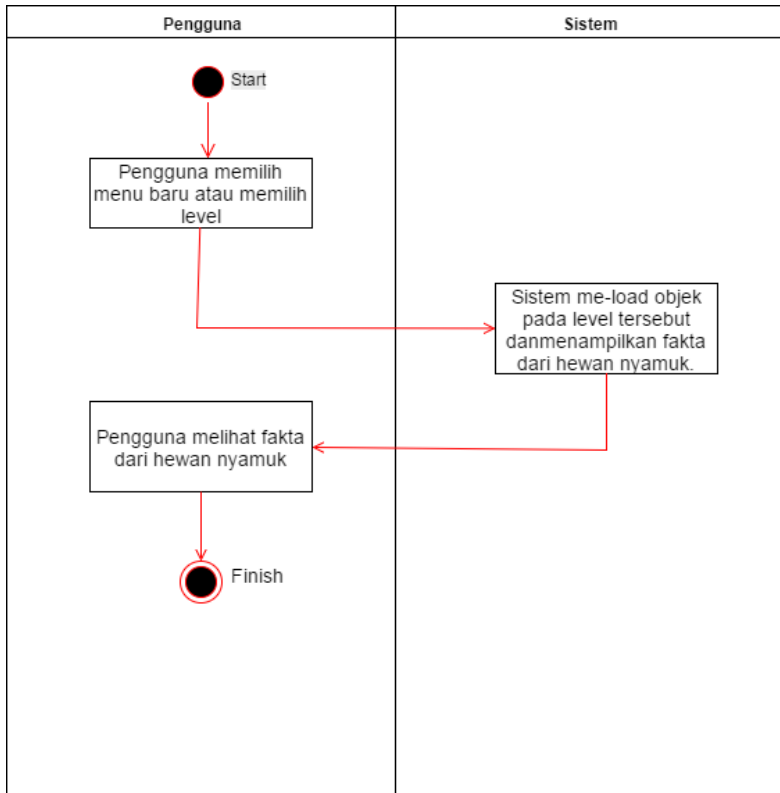
Diagram Aktifitas Memilih Menu Navigasi dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Aktivitas Memilih Menu Navigasi

3.4.2.2.2 Diagram Aktifitas Melihat Info Fakta (*Fun Fact*) dari Hewan Nyamuk

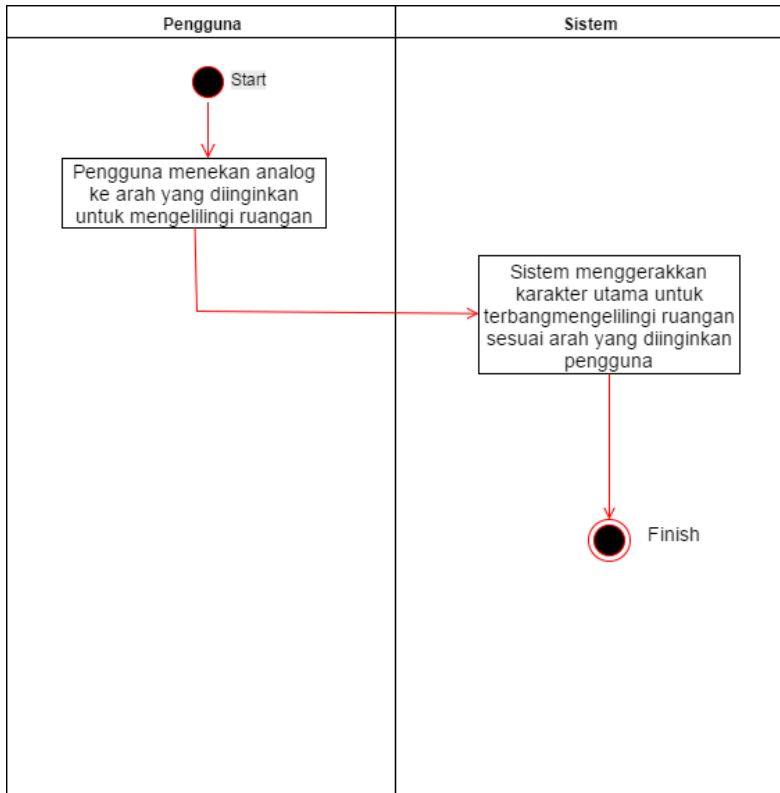
Diagram Aktifitas melihat info fakta (*fun fact*) dari hewan nyamuk dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Aktivitas Melihat Info Fakta

3.4.2.2.3 Diagram Aktifitas Bergerak atau Terbang Mengelilingi Ruangan

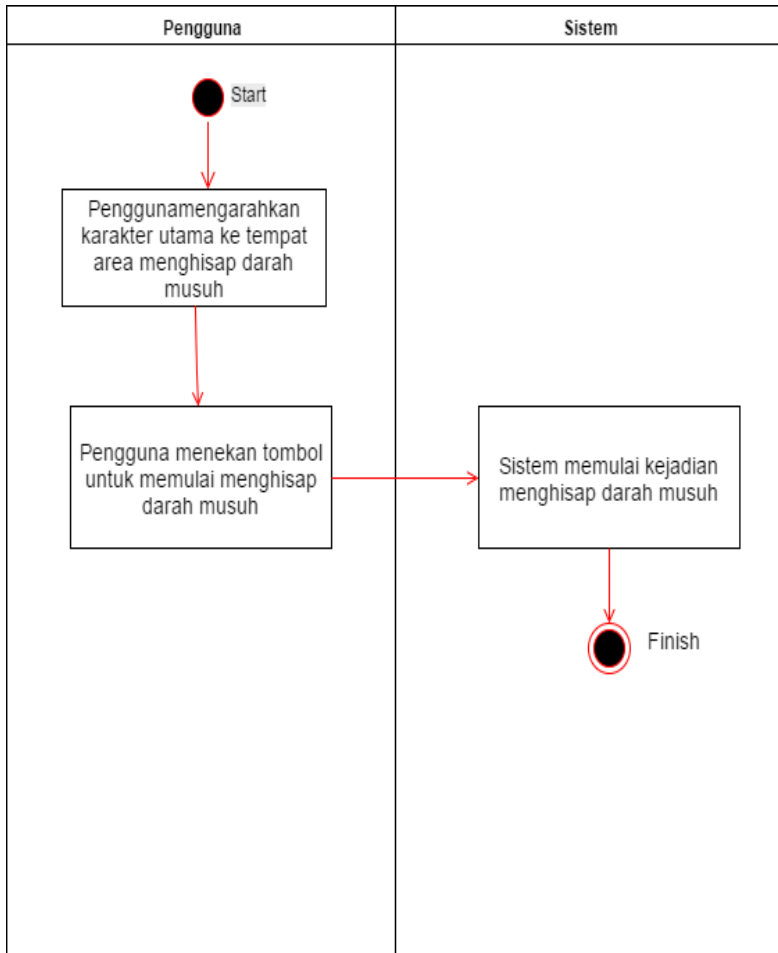
Diagram Aktifitas bergerak atau terbang mengelilingi ruangan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Aktivitas Bergerak atau Terbang Mengelilingi Ruangan

3.4.2.2.4 Diagram Aktifitas Menghisap Darah Musuh atau Target

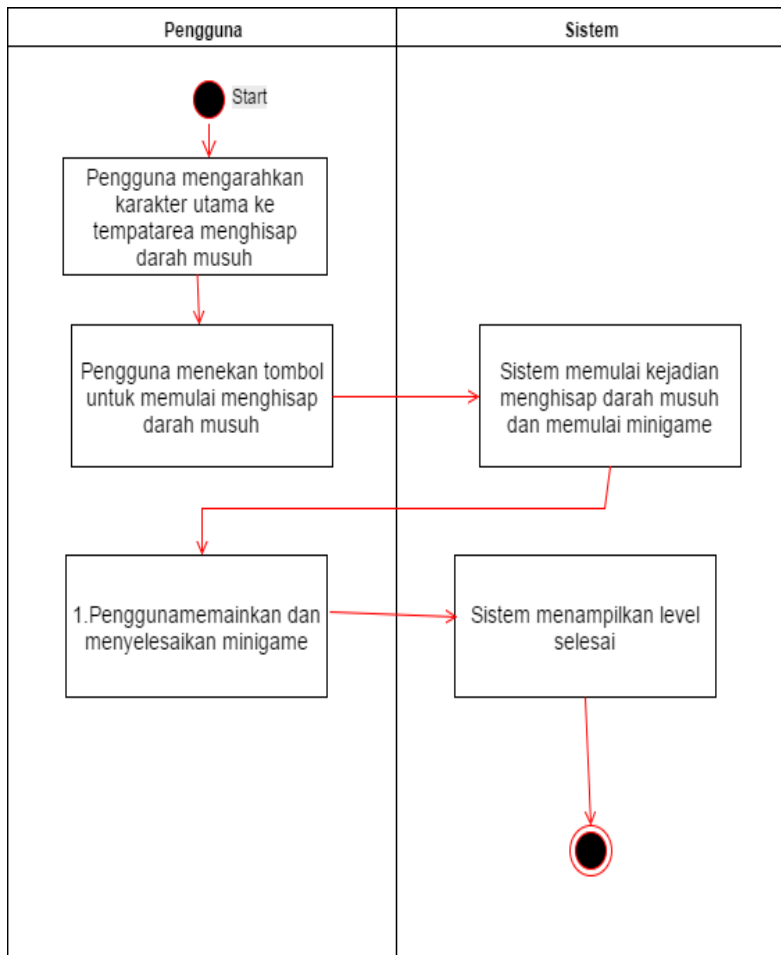
Diagram Aktifitas menghisap darah musuh atau target dapat dilihat pada Gambar 3.5.



Gambar 3.5 Menghisap Darah Musuh atau Target

3.4.2.2.5 Diagram Aktifitas Menyelesaikan *minigame*

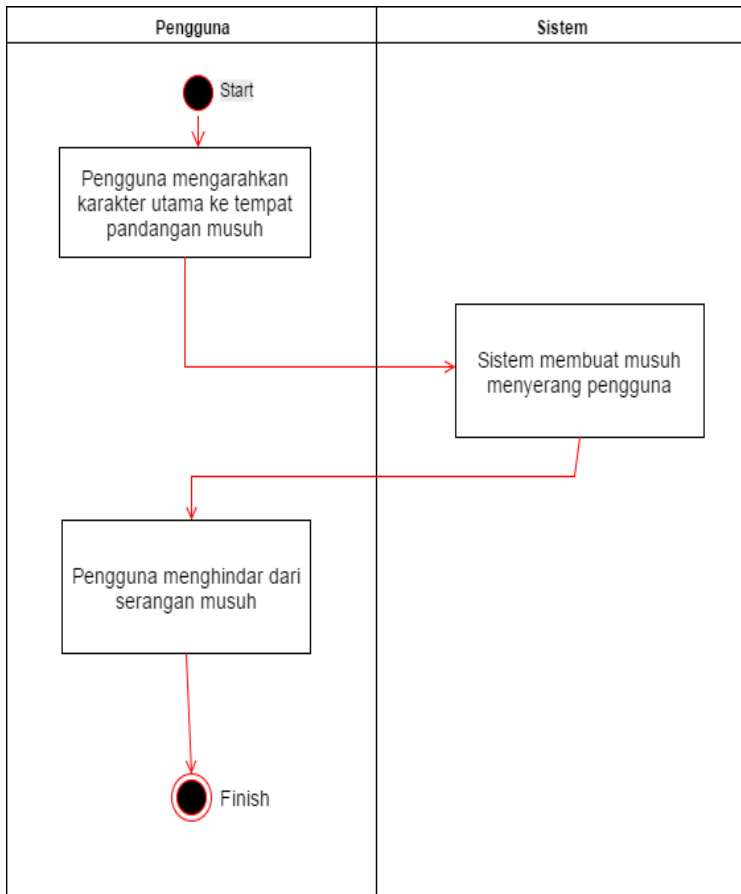
Diagram Aktifitas menyelesaikan *minigame* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Aktivitas Menyelesaikan *Minigame*

3.4.2.2.6 Diagram Aktifitas Menghindari Pukulan Musuh atau Target

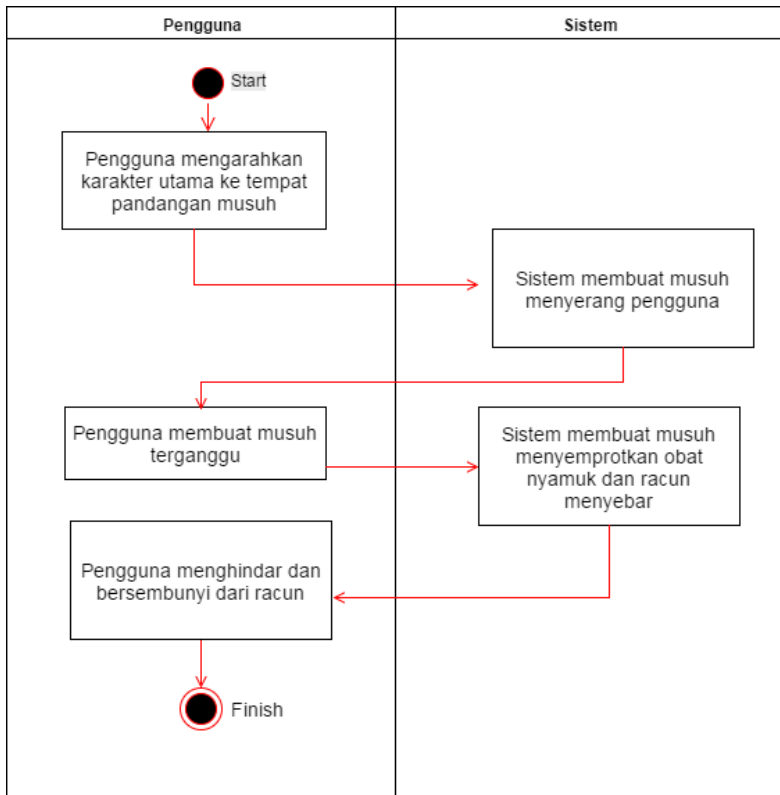
Diagram Aktifitas menghindari pukulan musuh atau target dapat dilihat pada Gambar 3.7.



Gambar 3.7 Diagram Aktivitas Menghindari Pukulan Musuh atau Target

3.4.2.2.7 Diagram Aktifitas Bersembunyi dari Racun

Diagram Aktifitas bersembunyi dari racun dapat dilihat pada Gambar 3.8.



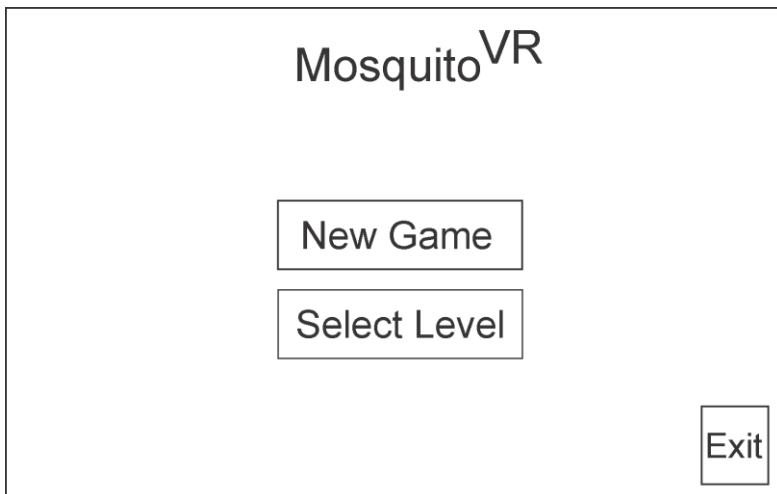
Gambar 3.8 Diagram Aktivitas Bersembunyi dari Racun

3.4.3 Perancangan Antarmuka Pengguna

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan jendela tampilan. Dalam aplikasi ini terdapat beberapa tampilan antarmuka, yaitu **Antarmuka Menu Utama**, **Antarmuka Layar Tunggu (Loading Screen)**, **Antarmuka Dalam Permainan (Ingame)**, dan **Antarmuka Minigame**.

3.4.3.1 Antarmuka Menu Utama

Antarmuka menu utama merupakan tampilan pertama dari permainan saat permainan pertama kali dijalankan. Tampilan ini memiliki 4 tombol, yaitu tombol “**New Game**” untuk memulai permainan, “**Select Level**” memilih level permainan, ikon gerigi untuk mengubah pengaturan dalam permainan, dan ikon pintu untuk keluar dari permainan. Desain antarmuka ini ditunjukkan pada Gambar 3.9.

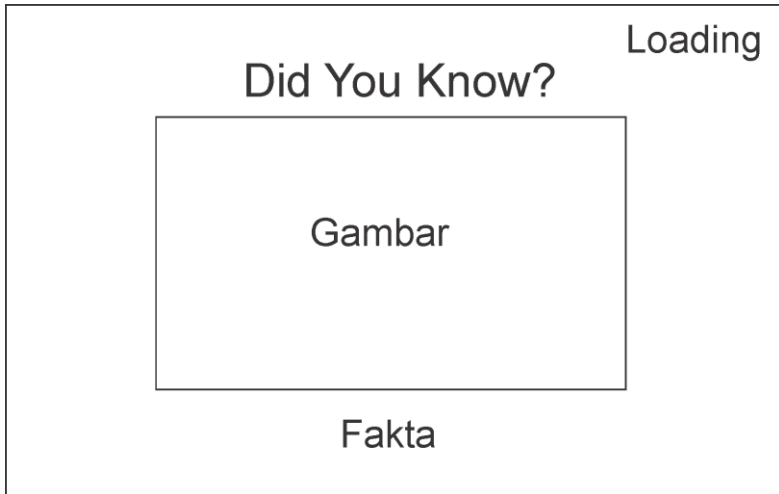


Gambar 3.9 Rancangan Antarmuka Menu Utama Permainan

3.4.3.2 Antarmuka Layar Tunggu (*Loading Screen*)

Antarmuka Layar Tunggu (*Loading Screen*) merupakan antarmuka ketika pengguna memilih memulai permainan atau memilih level permainan. Antarmuka ini berisikan waktu tunggu untuk masuk permainan, fakta menarik tentang hewan nyamuk, dan persiapan untuk memasuki permainan ketika permainan sudah me-load level yang diinginkan. Di antarmuka ini terdapat fakta-fakta menarik tentang hewan nyamuk (*funfact*) yang dipilih

secara acak dan ditampilkan saat permainan dalam proses menyiapkan level. Setiap masuk ke layar tunggu, fakta yang diambil sejumlah satu fakta, dan akan berubah apabila memasuki layar tunggu lagi pada saat yang berbeda. Tampilan ini ditunjukkan pada Gambar 3.10.

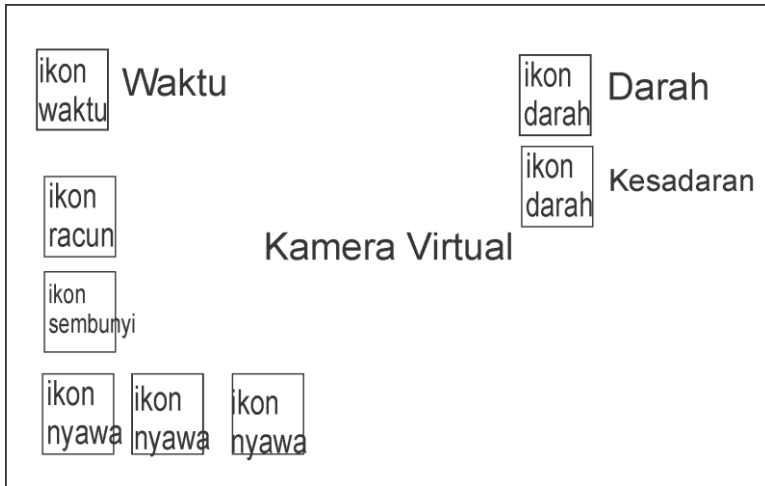


Gambar 3.10 Rancangan Antarmuka Layar Tunggu (*Loading Screen*)

3.4.3.3 Antarmuka dalam Permainan (*Ingame*)

Antarmuka dalam permainan (*Ingame*) merupakan tampilan antarmuka yang ditampilkan ketika pengguna sudah memasuki permainan utama. Antarmuka ini adalah antarmuka utama dari permainan. Pada antarmuka ini pengguna dapat melihat lingkungan dan ikon-ikon yang menunjukkan status dalam permainan. Ikon-ikon yang ditunjukkan adalah ikon nyawa, waktu, darah, sembunyi, dan racun. Antarmuka ini akan dipasang pada objek karakter utama atau kamera virtual. Sehingga ketika karakter pengguna bergerak, antarmuka ini akan

mengikuti karakter pengguna. Tampilan ini ditunjukkan pada Gambar 3.7.



**Gambar 3.11 Rancangan Antarmuka Dalam Permainan
(Ingame)**

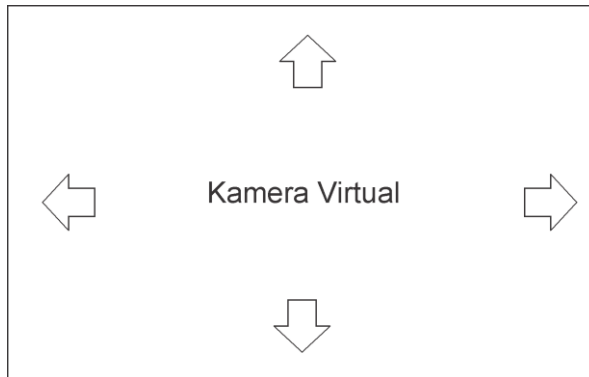
Penjelasan dari antarmuka dalam permainan, yang ditunjukkan sesuai dengan nomornya, yaitu sebagai berikut:

1. **Time**, merupakan antarmuka yang menunjukkan waktu yang tersisa untuk menyelesaikan level
2. **Blood**, merupakan antarmuka yang menunjukkan berapa darah target yang sudah dihisap
3. **Alert**, merupakan antarmuka yang menunjukkan berapa kesadaran target atau musuh dengan keberadaan pengguna
4. **Poison**, merupakan antarmuka yang menunjukkan adanya racun di dalam ruangan
5. **Safe Zone**, merupakan antarmuka yang menunjukkan berada dalam tempat aman dimana dapat mengurangi kesadaran target dan menghindari dari racun

6. **Health**, merupakan antarmuka yang menunjukkan nyawa pengguna

3.4.3.4 Antarmuka *Minigame*

Antarmuka *minigame* merupakan tampilan antarmuka yang ditampilkan ketika pengguna berada di area tempat menghisap darah pada target dan pengguna memulai menghisap darah. Antarmuka ini terdiri dari 4 ikon arah yang harus ditekan ketika muncul untuk mendapatkan darah dari target. Tampilan ini ditunjukkan pada Gambar 3.8.



Gambar 3.12 Rancangan Antarmuka *Minigame*

3.4.4 Perancangan Alur Permainan

Permainan yang dibuat pada tugas akhir ini memiliki alur permainan yang dimulai ketika pengguna menjalankan permainan, untuk selengkapnya alur pada permainan ini yaitu antara lain:

1. Saat permainan dijalankan maka pemain akan melihat halaman utama. Seperti yang telah dijelaskan pada subbab 3.4.3.1, halaman utama memiliki tiga tombol aksi yaitu, tombol *New Game*, *Load Level*, dan tombol *Exit*.

- a. Apabila memilih tombol *New Game*, sistem akan menampilkan antarmuka tunggu saat sistem mempersiapkan level.
 - b. Apabila memilih tombol *Load Level*, sistem akan menampilkan tombol lagi untuk memilih level yang ingin dimainkan, setelah memilih level, sistem akan menampilkan antarmuka tunggu saat sistem mempersiapkan level.
2. Setelah memilih tombol *New Game* atau memilih level yang ingin dimainkan, maka pengguna akan menuju ke antarmuka layar tunggu ketika sistem mempersiapkan level. Penjelasan mengenai antarmuka layar tunggu seperti yang telah dijelaskan pada subbab 3.4.3.2.
3. Setelah sistem mempersiapkan level, Pengguna kemudian akan ditampilkan *scene* permainan serta antarmuka dalam permainan (*ingame*). Pada saat ini pemain mulai memainkan dengan menggerakkan karakter utama untuk menghisap darah dari target yang ada. Ikon dan angka status permainan pada antarmuka dalam permainan dijelaskan pada subbab 3.4.3.3.
4. Pengguna harus terbang menuju tempat area menghisap darah pada target yaitu objek manusia.
5. Ketika pengguna berada di area pandangan target selama beberapa detik, target atau objek manusia akan berusaha memukul pengguna dan kesadaran target akan bertambah.
6. Apabila pengguna terkena pukulan, nyawa pengguna akan berkurang.
7. Ketika pemain sampai pada area untuk menghisap darah, maka akan muncul notifikasi untuk menekan tombol pada kontroller untuk memulai *minigame* menghisap darah target.
8. Pengguna akan ditampilkan antarmuka *minigame* untuk menghisap darah target. Untuk memainkan *minigame*, pengguna harus menekan tombol arah pada kontroller ketika muncul ikon arah pada antarmuka sesuai dengan ikon arah yang muncul seperti yang dijelaskan pada subbab 3.4.3.4.

9. Ketika berhasil menekan tombol arah yang benar, maka darah yang terhisap akan bertambah dan terlihat pada antarmuka dalam permainan. Apabila menekan tombol yang salah, maka kesadaran akan bertambah dan terlihat pada antarmuka dalam permainan.
10. Apabila darah yang terhisap mencapai maksimal, maka permainan dikatakan menang dan dapat melanjutkan ke level selanjutnya.
11. Apabila kesadaran mencapai maksimal dan pengguna masih berada di area menghisap darah, maka objek manusia akan memukul pengguna dan permainan dinyatakan kalah dan harus mengulangi level dari awal.
12. Apabila kesadaran mencapai maksimal dan pengguna berada di luar area menghisap darah, maka objek manusia akan menyemprotkan obat nyamuk untuk menyebarkan racun.
13. Apabila racun tersebar, maka ikon racun akan muncul di antarmuka dalam permainan (*ingame*). Apabila pengguna berada di area racun, maka nyawa pemain akan berkurang terus menerus dalam waktu beberapa detik. Pengguna harus mencari area bersembunyi untuk tidak terkena racun.
14. Apabila pengguna berada dalam area bersembunyi, maka ikon pelindung akan muncul di antarmuka dalam permainan (*ingame*) dan pengguna dapat bersembunyi dari racun dan mengurangi kesadaran manusia
15. Apabila nyawa pengguna habis karena terkena racun atau terkena pukulan, maka permainan dikatakan kalah dan harus mengulangi level dari awal.
16. Permainan dikatakan menang apabila semua level sudah berhasil memenuhi target.

3.4.5 Perancangan Aturan Permainan

Dalam permainan yang akan dibuat pada tugas akhir ini, terdapat beberapa aturan yaitu sebagai berikut:

1. Pada awal level permainan, pengguna akan diberikan tiga nyawa dalam menyelesaikan satu level. Pengguna juga

diberikan waktu sebanyak 200 detik untuk menyelesaikan satu level dengan nilai darah yang dihisap dan kesadaran dimulai dari nol.

2. Nyawa pengguna dan waktu permainan tidak dapat bertambah apabila sudah berkurang.
3. Nyawa pengguna akan berkurang sebanyak setengah selama beberapa detik apabila berada di area racun, dan berkurang sebanyak satu nyawa apabila terkena pukulan dari target.
4. Darah yang terhisap bertambah sebanyak sepuluh (10) dan kesadaran bertambah sebanyak lima (5) apabila pengguna berhasil menekan tombol arah yang benar di dalam *minigame* dan kesadaran bertambah sebanyak dua puluh (20) apabila pengguna menekan tombol arah yang salah di dalam *minigame*.
5. Nilai maksimal dari darah yang terhisap dan kesadaran manusia sebesar seratus (100).
6. Semua level dalam permainan sudah terbuka dari awal dan dapat dimainkan tanpa harus memainkan level selanjutnya.
7. Apabila darah yang terhisap mencapai nilai maksimal, maka level tersebut dinyatakan berhasil.
8. Apabila nyawa pengguna habis, maka level tersebut dinyatakan gagal dan harus mengulangi level yang dimainkan.
9. Permainan dikatakan menang apabila sudah menyelesaikan target pada semua level.

3.4.6 Perancangan Skenario Level Permainan

Dalam permainan yang dibuat pada tugas akhir ini, terdapat tiga level dengan ruangan yang berisi objek manusia dengan melakukan aktifitas yang berbeda-beda. Level 1 berada di ruangan tempat tidur orang tua dimana objek manusianya yaitu seorang ayah yang sedang melakukan aktifitas tidur, pada level 2 berada di ruangan kamar tidur anak laki-laki dengan objek manusianya yaitu anak laki-laki yang sedang melakukan aktifitas mengetik komputer, dan pada level 3 berada di ruangan keluarga

dengan objek manusianya yaitu anak perempuan yang sedang melakukan aktifitas melihat televisi. Level yang terdapat pada permainan dapat dimainkan secara berurutan maupun melompati level sebelumnya. Untuk melanjutkan ke level selanjutnya, pengguna harus menghisap darah sampai kapasitas darah mencapai 100 dan berhasil menghisap darah di semua daerah hisap pada manusia.

Skenario untuk tiap level akan dibuat secara manual tetapi akan mengikuti aturan tertentu agar tingkat kesulitan berawal dari mudah yang berangsur-angsur menjadi lebih susah. Untuk jumlah penambahan atau pengurangan nyawa, kesadaran, atau darah disamakan pada semua level. Aturan skenario level ditunjukkan pada Tabel 3.10.

Tabel 3.10 Skenario Level Permainan

Level/ Stage	Ruangan	Manusia dan Aktifitasnya	Rintangan	Jumlah daerah target
1	Kamar Tidur Orang Tua	Ayah; Tidur	Pukulan	1
2	Kamar Tidur Anak Laki-laki	Anak Laki- Laki;Tidur	Pukulan, Usiran ketika berada di area pandangan dan Racun	1
3	Ruang Keluarga	Anak Perempuan; Melihat Televisi	Pukulan, Usiran ketika berada di area pandangan dan Racun	2

3.4.7 Perancangan Kontrol Permainan

Kontrol dalam permainan ini yaitu menggunakan *google cardboard* dan *bluetooth controller*. *Google cardboard* digunakan untuk menggerakkan posisi dan rotasi pandangan pengguna dan karakter utama dalam game terhadap lingkungan virtual, pengguna cukup menggerakkan kepala untuk melihat lingkungan dalam permainan kearah yang diinginkan. Untuk *bluetooth controller* digunakan untuk memberi perintah terhadap karakter utama seperti terbang, menghisap darah, atau menghentikan permainan. Tombol analog digunakan untuk bergerak maju, mundur, ke kiri dan ke kanan. Tombol A atau X digunakan untuk memulai menghisap darah apabila sudah berada di area hisap darah target. Tombol Start digunakan untuk menghentikan permainan.

(Halaman Ini Sengaja dikosongkan)

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas tentang implementasi dari perancangan permainan realitas virtual kehidupan hewan nyamuk dari sudut pandang orang pertama menggunakan *google cardboard*. Di dalamnya mencakup proses penerapan dan pengimplementasian sistem dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi dari tugas akhir ditunjukkan pada Tabel 4.1 dan Tabel 4.2 sebagai lingkungan pembangunan aplikasi.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak (1)

Perangkat Keras	Prosesor : - Intel(R) Core(TM) i7-4700MQ CPU @2.4ghz Quad-Core Memori : - 8192 MB RAM
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 64-bit Perangkat Pengembang : - Unity3D - Microsoft Visual Studio Community 2013 - Adobe Photoshop CS6 Trial - Blender 2.78

Tabel 4.2 Lingkungan Implementasi Perangkat Lunak (2)

Perangkat Keras	Prosesor : - Intel(R) Core(TM) i5-M460 CPU @2.53ghz Dual-Core
-----------------	---

	Memori : - 2048 MB RAM
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 7 32-bit Perangkat Pengembang : - Unity3D - MonoDevelop - Adobe Photoshop CS6 Trial - Blender 2.78

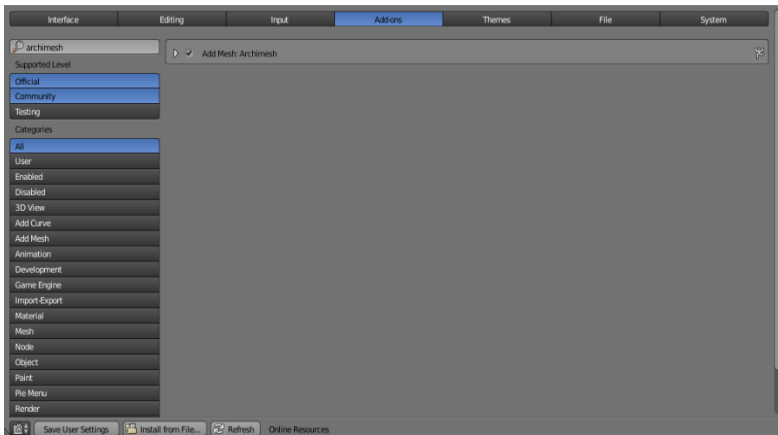
4.2 Implementasi Lingkungan Ruangan Virtual

Pada tahap ini, hal yang pertama kali dilakukan adalah membuat objek ruangan dan objek pendukungnya dengan menggunakan aplikasi Blender. Untuk memudahkan pembuatan ruangan, penulis menggunakan sebuah *add-on* yaitu Archimesh. Archimesh adalah *add-on* yang disediakan aplikasi Blender untuk memudahkan dalam pembuatan objek ruangan dan objek pendukungnya. Untuk menambahkan *add-on*, dapat dilakukan dengan mengklik menu “File” lalu memilih “User Preferences”. Kemudian dilanjutkan dengan memilih tab “Add-ons” dan muncul tampilan seperti pada Gambar 4.1.



Gambar 4.1 Tampilan “User Preferences” sebelum terinstall *add-on* Archimesh

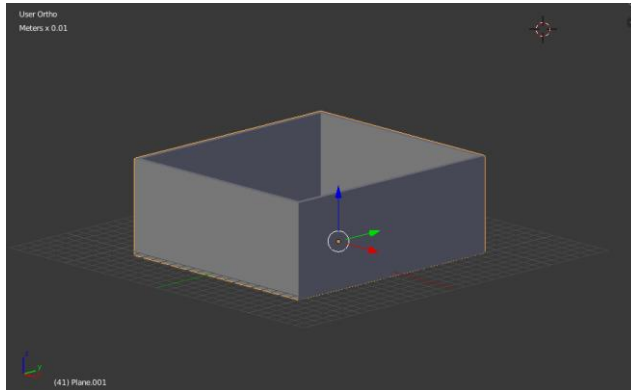
Langkah berikutnya adalah mencari *add-on* dengan cara mengetikkan pada kolom pencarian di pojok kiri atas dengan kata “archimesh”, maka akan keluar hasil pencarian. Setelah itu klik centang pada hasil pencarian tersebut. Apabila sudah tercentang, maka *add-on* Archimesh telah terinstall seperti yang terlihat pada Gambar 4.2.



Gambar 4.2 Tampilan “User Preferences” setelah terinstall *add-on* Archimesh

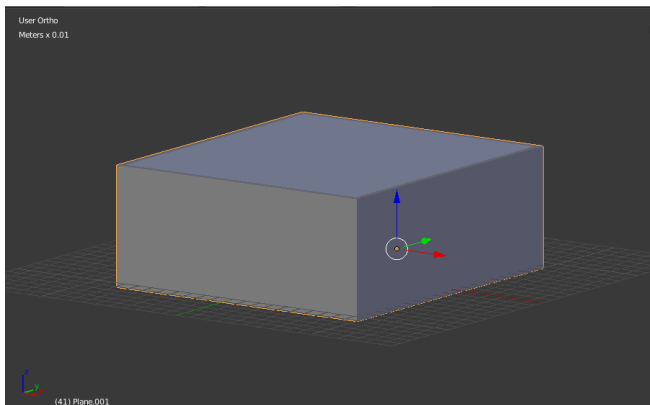
4.2.1 Implementasi Ruang Virtual

Setelah *add-on* Archimesh terpasang, langkah berikutnya adalah membuat objek ruangan pada blender. Ruang yang dibuat adalah ruangan yang lingkungan permainan dan pembatas zona permainan yang dibuat. Ruang pertama yang dibuat adalah ruangan tempat tidur. Langkah pertama adalah menambahkan object kubus pada layar kerja Blender, lalu diatur panjang, lebar, dan tingginya sedemikian rupa untuk membuat dinding dari suatu ruangan, setelah objek dinding terbentuk, objek tersebut akan di-*extrude* sedemikian rupa sehingga terbentuk kerangka ruangan berupa dinding tanpa atap dan lantai seperti yang terlihat pada Gambar 4.3.



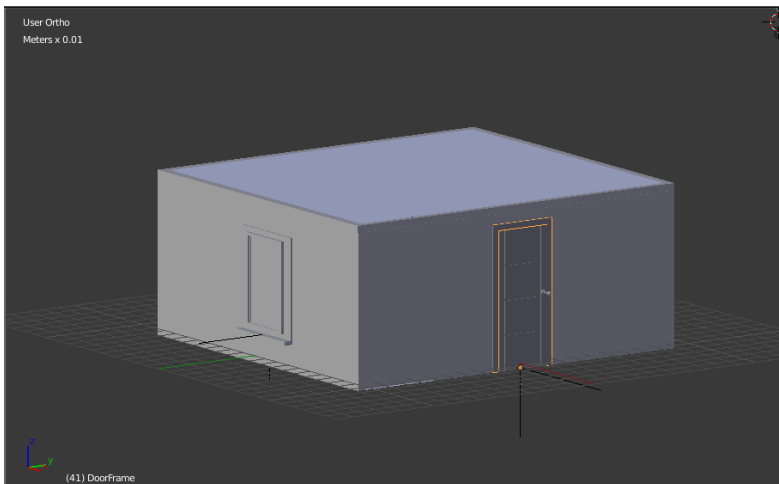
Gambar 4.3 Hasil *Extrude* dari Dinding dan Terbentuknya Kerangka Ruang

Selanjutnya adalah membuat lantai dan atap dengan cara mengklik setiap rusuk dari lantai dan atap lalu tekan tombol “F” pada *keyboard* untuk melakukan perintah membuat sisi dari rusuk lantai dan atap sehingga lantai dan atap terbentuk. Dan akhirnya terbentuk ruangan seperti yang terlihat pada Gambar 4.4.



Gambar 4.4 Hasil Bentuk Ruang setelah Penambahan Atap dan Lantai

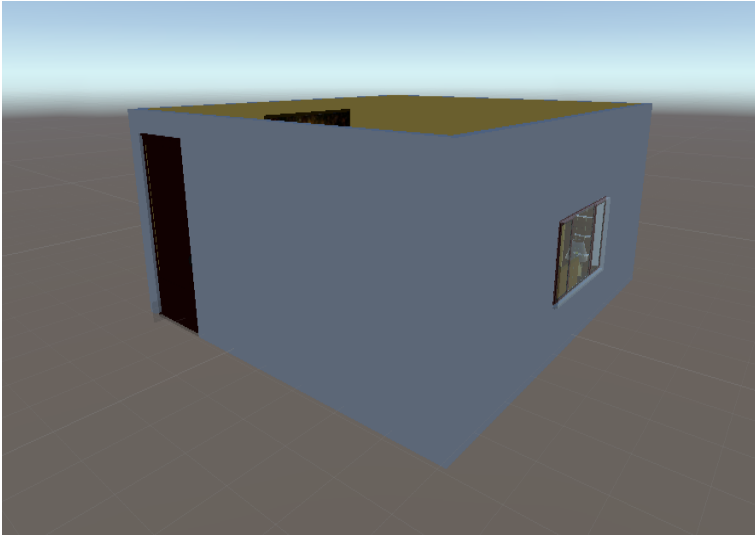
Setelah objek ruangan telah terbentuk, ditambahkan objek-objek pendukung ruangan seperti jendela dan pintu. Objek tersebut didapat dari *add-on* Archimesh. Setelah objek pintu dan jendela ditambahkan, objek pintu dan jendela dilakukan pengaturan bentuk model, ketinggian, jumlah, dan ketebalannya. Selanjutnya pintu dan jendela diletakkan pada dinding ruangan yang diinginkan. Hasil dari pemasangan jendela dan pintu dapat dilihat pada Gambar 4.5.



Gambar 4.5 Hasil Akhir Bentuk Ruangan

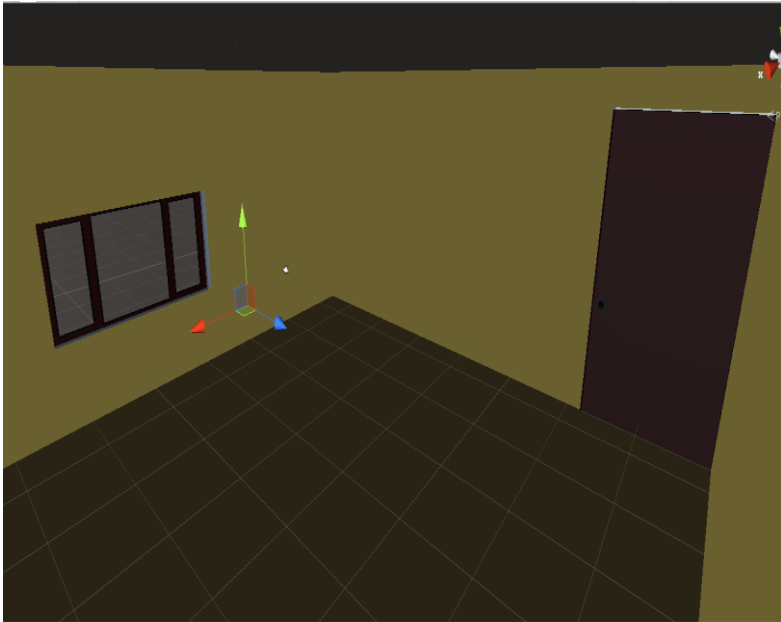
Proses selanjutnya ialah memberi material pada sisi dinding ruangan dan objek pendukung ruangan yang diinginkan untuk memberi warna pada objek tersebut. Material objek tersebut tidak perlu diatur di aplikasi Blender karena akan diatur di aplikasi Unity3D. Setelah semua objek diberi material, objek diekspor dengan ekstensi “.fbx” agar dapat diimpor ke aplikasi Unity dengan cara klik menu “File” lalu pilih “Export” lalu klik “FBX (.fbx)”, selanjutnya centang “Selected Object” pada kiri bawah lalu pilih lokasi untuk menyimpan objek, lalu klik “Export FBX”. Setelah objek diekspor, langkah berikutnya adalah

memasukkan file objek ruangan tersebut ke dalam Unity3D dengan cara drag file “.fbx” yang telah dibuat pada folder “Assets” di *project* Unity yang telah terbuka. Kemudian drag *asset* yang telah ter-import pada *Scene* untuk membuat lingkungan virtual pada unity. Tampilan ruangan setelah dimasukkan ke dalam Unity terlihat pada Gambar 4.6.



Gambar 4.6 Tampilan Ruangan Dalam Aplikasi Unity

Selanjutnya adalah pemberian warna pada material objek di unity dengan cara klik objek yang telah dimasukkan, lalu klik material di panel *inspector*, lalu klik *base color* di bagian material untuk mengubah warna objek. Setelah mengubah warna objek, dilanjutkan dengan member pencahayaan pada mall dengan mengklik menu “GameObject”>”Light”. Pada permainan ini jenis pencahayaan yang digunakan adalah satu *Directional Light*, dengan penambahan *Point Light* untuk member kesan lampu ruangan. Tampilan ruangan setelah diberi warna dan lampu terlihat pada Gambar 4.7.

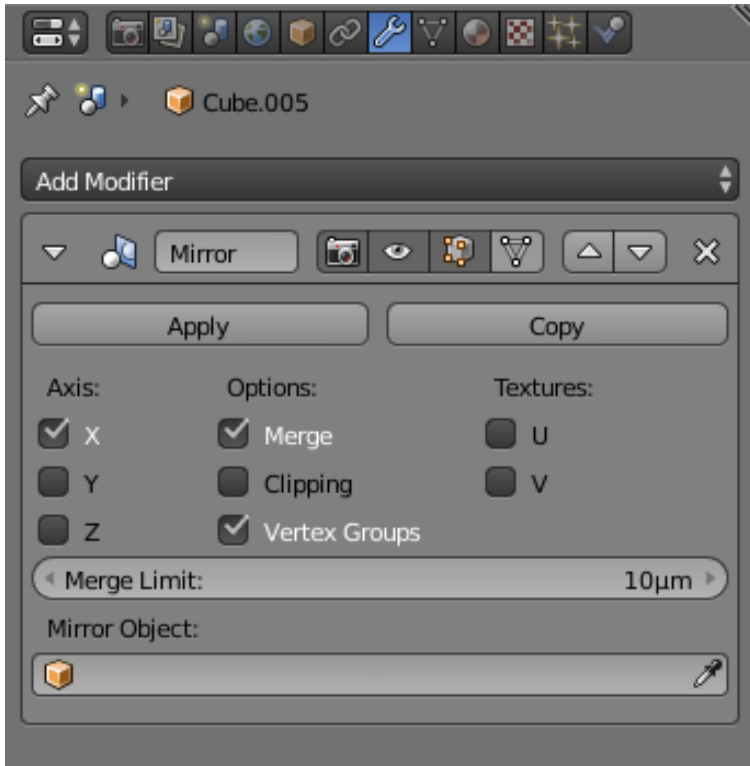


Gambar 4.7 Tampilan Ruangan Setelah Diberi Warna dan Pencahayaan

4.2.2 Implementasi Objek Pendukung Ruangan Virtual

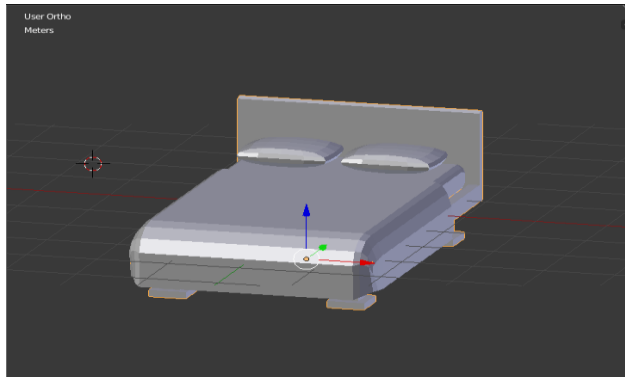
Setelah Ruangan Virtual terbentuk dan dimasukkan ke dalam aplikasi Unity, langkah berikutnya adalah membuat objek pendukung ruangan atau properti pada blender. Dari pembuatan tipe ruangan sebelumnya, maka objek pendukung pertama yang dibuat adalah tempat tidur. Langkah pertama adalah membuat penyangga tempat tidur terlebih dahulu dengan menambahkan object kubus pada layar kerja Blender, lalu diatur panjang, lebar, dan tingginya untuk membuat penyangga tempat tidur. lalu objek diberi *modifier* yang disebut *mirror* untuk membuat objek menjadi dua objek yang simetris dengan cara klik tab “*modifier*” pada tab “*Properties*”, lalu klik “*Add Modifier*”, pilih “*Mirror*”.

Tampilan setelah diberi *modifier* berupa *mirror* terlihat seperti Gambar 4.8.



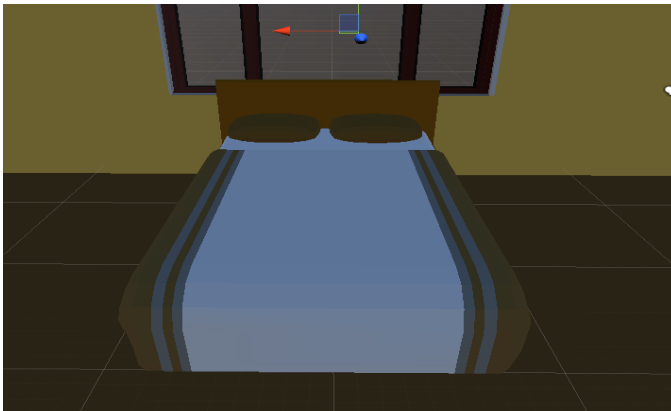
Gambar 4.8 Mirror Modifier pada Objek Penyangga Tempat Tidur

Setelah diberi *modifier*, objek di-*extrude*, diedit, dimodifikasi sehingga membentuk penyangga tempat tidur. Selanjutnya membuat objek matras kasur dan bantal dengan cara yang sama dengan pembuatan penyangga tempat tidur, lalu diletakkan diatas penyangga tempat tidur dan menghasilkan objek tempat tidur. Sehingga terlihat seperti pada Gambar 4.9.



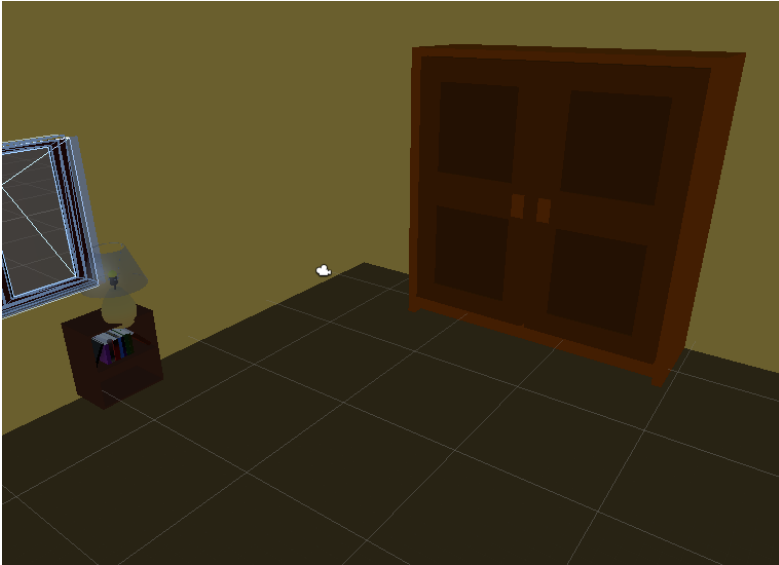
Gambar 4.9 Hasil Akhir Objek Tempat Tidur

Selanjutnya yaitu pemberian material setiap bagian dari tempat tidur agar dapat diberi warna dengan cara sama seperti pembuatan ruangan sebelumnya. Setelah pemberian material, objek tempat tidur diekspor dalam bentuk ekstensi “.fbx”. Setelah itu objek dimasukkan ke dalam aplikasi Unity dan diletakkan pada *scene* Unity dan diletakkan ke dalam objek ruangan. Lalu diberikan warna pada material sehingga terlihat seperti pada Gambar 4.10.



Gambar 4.10 Tampilan Tempat Tidur Dalam Aplikasi Unity

Selain objek tempat tidur, objek pendukung lain yang dibuat yaitu rak buku, lampu duduk, dan lemari. Dengan cara yang sama, objek yang telah dibuat dimasukkan ke dalam *scene unity* dan dimasukkan ke dalam objek ruangan. Tampilan ruangan setelah penambahan kursi, lampu dan lemari akan terlihat seperti pada Gambar 4.11.

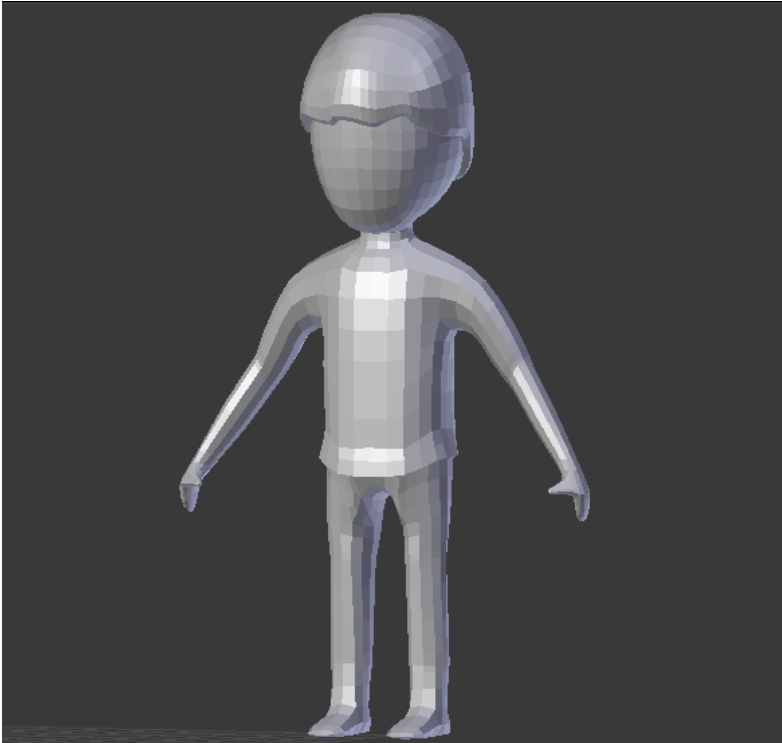


Gambar 4.11 Tampilan Rak Buku, Lampu Duduk dan Lemari Dalam Aplikasi Unity

4.3 Implementasi Objek Manusia

Setelah Ruangan Virtual dan objek pendukungnya terbentuk dan dimasukkan ke dalam aplikasi Unity, langkah berikutnya adalah membuat objek target dari permainan atau penghuni ruangan virtual tersebut yaitu manusia. Objek manusia yang pertama kali dibuat adalah seorang laki-laki. Cara pembuatannya sama seperti pembuatan objek pendukung, objek pertama yang dibuat adalah kubus, lalu diberi *modifier mirror*

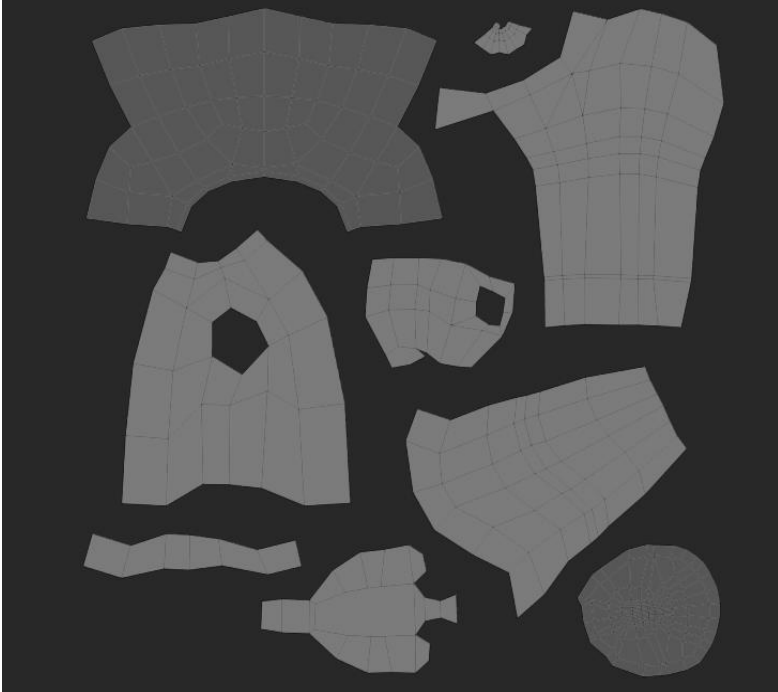
untuk membuat objek menjadi dua objek yang simetris, lalu dibentuk sedemikian rupa sehingga membentuk manusia. Untuk membentuk rambutnya, *modifier mirror* yang ditambahkan harus di-*apply* terlebih dahulu agar pembentukan objek selanjutnya dapat dibentuk asimetris. Tampilan manusia yang telah terbentuk akan terlihat seperti pada Gambar 4.12.



Gambar 4.12 Tampilan Objek Manusia

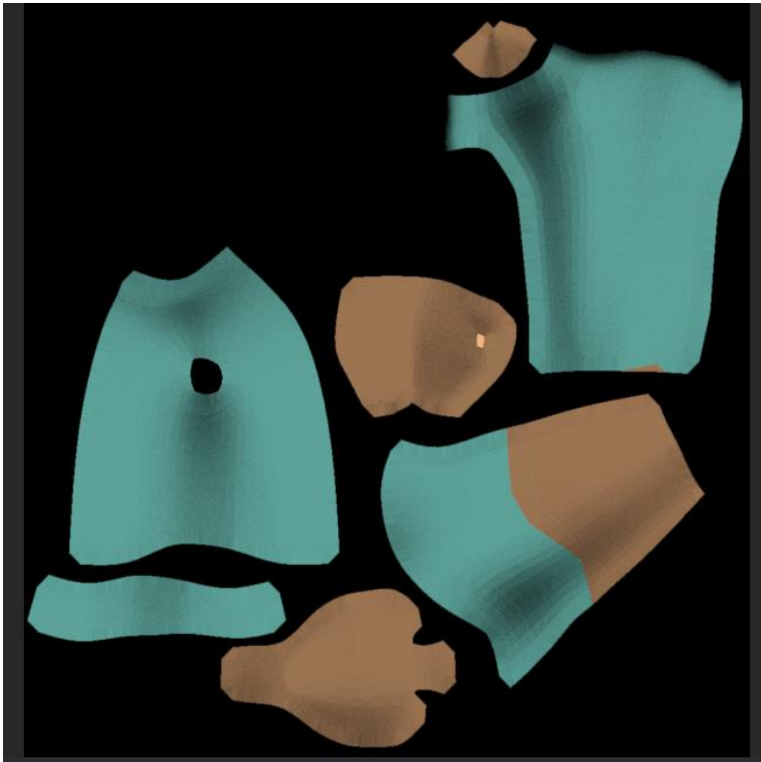
Setelah objek manusia terbentuk, objek manusia perlu dilakukan *UV Mapping* untuk membuat layout dari sisi objek manusia untuk diberi tekstur. cara melakukannya yaitu blok semua sisi manusia, lalu tekan “U” pada *Keyboard*, lalu klik

“*Smart UV Project*”, maka layout UV akan terlihat seperti pada Gambar 4.13.



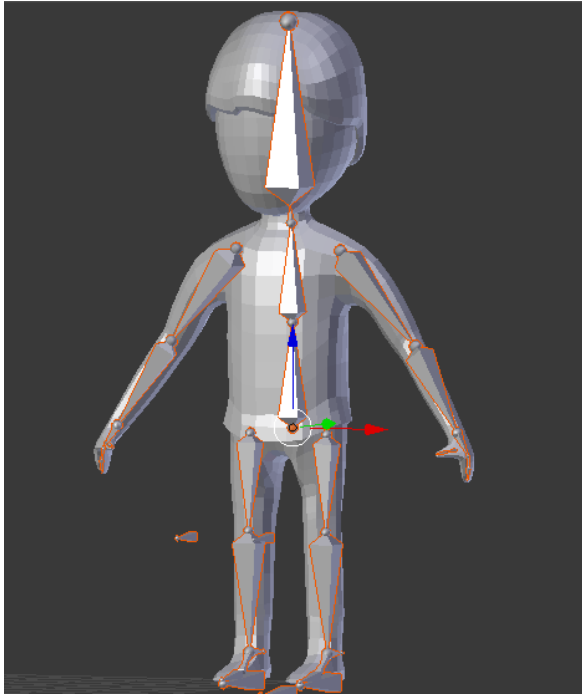
Gambar 4.13 Hasil *UV Mapping* dari Objek Manusia

Setelah layout UV telah diproses, simpan file layout dengan cara klik menu “*UVs*” pada layar kerja “*UV/Image Editor*”, lalu klik “*Export UV Layout*” lalu pilih lokasi untuk simpan layout lalu klik “*Export UV*”. Selanjutnya file layout UV yang tersimpan dibuka menggunakan aplikasi Adobe Photoshop dan diberikan warna pada layout UV sesuai dengan keinginan di setiap bagian objek manusia. Untuk daerah diluar layout UV dapat diberi warna hitam. Setelah selesai, hasil pewarnaan layout UV tadi disimpan dalam bentuk file baru. Hasil dari pewarnaan layout akan terlihat seperti pada Gambar 4.14.



Gambar 4.14 Hasil Pewarnaan UV Layout

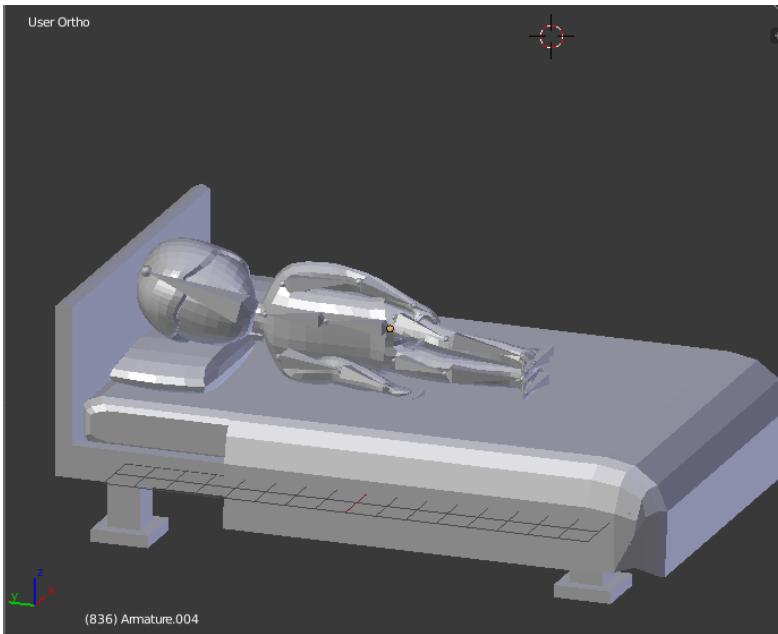
Proses selanjutnya yaitu proses *rigging* pada objek manusia. *Rigging* adalah proses penambahan kerangka tulang pada objek agar dapat digerakkan seperti manusia aslinya. Cara penambahan kerangka yaitu dengan cara klik menu “Add”, lalu pilih “Armature”, lalu klik “Single Bone”, maka kerangka tulang akan ditambahkan. Kerangka tulang tersebut diatur sesuai posisi objek manusia lalu di-*extrude* sesuai dengan panjang bagian tubuh manusia seperti lengan, tangan, leher, kepala, lutut, kaki, dan lain-lain. Hasil akhir dari kerangka tulang yang ditambahkan pada objek manusia akan terlihat seperti pada Gambar 4.15.



Gambar 4.15 Hasil Kerangka Tulang pada Objek Manusia

Setelah penambahan kerangka tulang, selanjutnya adalah membuat animasi pada objek manusia. animasi yang pertama kali dibuat adalah animasi ketika tidur. Cara pembuatan animasi yaitu dengan cara menggerakkan kerangka tulang sesuai dengan aktivitas yang ingin dianimasikan. Cara membuat animasi yaitu pertama klik pada kerangka tulang yang sudah dibuat, lalu masuk ke dalam “*Pose Mode*”, lalu ubah posisi dan rotasi semua bagian kerangka tulang seperti melakukan aktivitas, dimana dalam animasi ini aktivitas yang dilakukan adalah tidur. Setelah itu, buka layar kerja baru lalu pilih “*Dope Sheet*”, lalu tekan “*New*” dan isi nama animasi yang dibuat, lalu tekan “*I*” pada *keyboard* untuk menyimpan pose kerangka tulang tadi ke dalam frame. Selanjutnya pindah ke frame selanjutnya dengan jarak tertentu

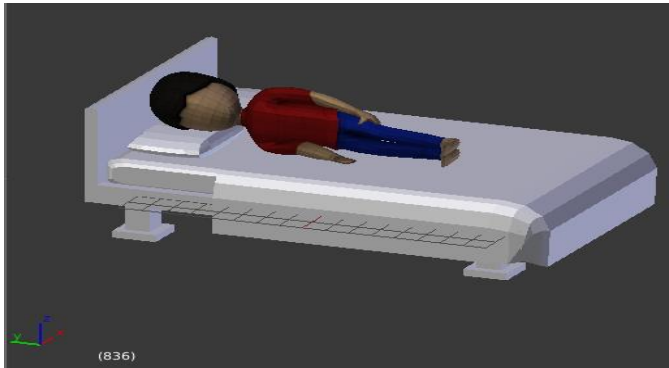
lalu ulangi langkah sebelumnya dengan pose yang sedikit berbeda agar animasi terlihat nyata. Hasil dari pembuatan animasi dapat terlihat seperti pada Gambar 4.16.



Gambar 4.16 Hasil Pembuatan Animasi pada Objek Manusia

Setelah itu lakukan pembuatan animasi lagi untuk aktifitas yang berbeda, seperti memukul nyamuk atau menggaruk bagian tubuh dengan cara sama dengan sebelumnya. Setelah pembuatan animasi selesai, objek manusia diekspor ke dalam bentuk fbx lalu dimasukkan ke dalam *scene* Unity ke dalam ruangan. Objek diposisikan sesuai dengan animasi aktifitas yang akan dilakukan. Setelah itu objek diberi tekstur pada *UV layout* yang sebelumnya dibuat dengan cara pindahkan file pewarnaan *UV Layout* yang dibuat ke dalam folder “Assets” Unity, lalu klik pada objek manusia, lalu pada bagian material, pilih *shader*

“Legacy Shader” > “Diffuse”, lalu pada “Base (RGB)”, klik “Select” dan pilih file pewarnaan layout UV yang sebelumnya dibuat dimana akan terlihat seperti pada Gambar 4.17. Hasil akhir setelah pengaturan posisi objek manusia dan penambahan tekstur akan terlihat pada Gambar 4.18.



Gambar 4.17 Penambahan Tekstur pada Objek Manusia



Gambar 4.18 Hasil Akhir Objek Manusia pada Aplikasi Unity

Untuk level permainan selanjutnya, terdapat objek manusia yang berbeda dengan aktifitas yang berbeda juga. Untuk level 2, objek manusia yang dibuat adalah anak laki-laki yang sedang melakukan aktifitas mengetik komputer. Dan untuk level 3, objek manusia yang dibuat adalah anak perempuan yang sedang melakukan aktifitas melihat televisi. Dengan cara pembuatan yang sama seperti sebelumnya, hasil pembuatan objek manusia untuk level 2 dan level 3 terlihat seperti pada Gambar 4.19 dan 4.20.



Gambar 4.19 Hasil Akhir Objek Manusia pada Level 2



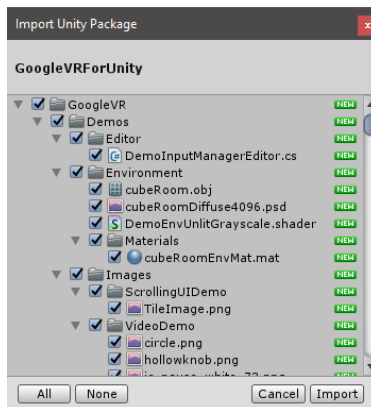
Gambar 4.20 Hasil Akhir Objek Manusia pada Level 3

4.4 Implementasi Objek Pengguna

Pada tahap ini, dilakukan pembuatan objek pengguna atau karakter utama dalam permainan yaitu nyamuk. Objek ini tidak memiliki model tiga dimensi, tetapi terdiri dari kamera realitas virtual, implementasi kontroller dengan penambahan *script* dan komponen *character controller*, dan penambahan komponen *collider* agar dapat mensimulasikan tabrakan dengan objek lain. untuk penjelasan lebih lanjut, akan dijelaskan pada subbab berikut.

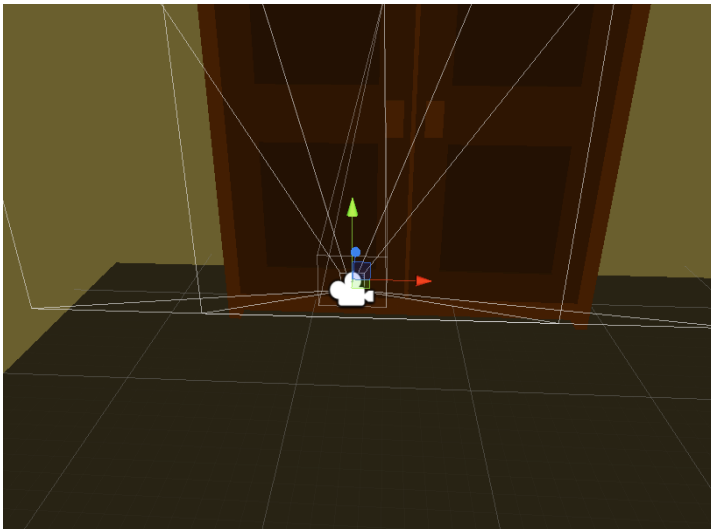
4.4.1 Implementasi Kamera Realitas Virtual

Yang pertama kali dilakukan pada tahap ini adalah mengintegrasikan *Google Cardboard SDK* dengan Unity agar pengguna dapat melihat lingkungan permainan menggunakan Google Cardboard. *Cardboard SDK* dapat diunduh pada situs <https://developers.google.com/vr/unity/>. Untuk mengimpor SDK ke dalam unity caranya yaitu klik menu “Assets”>”Import Packages”>”Custom Packages...” lalu pilih file *Cardboard SDK*, lalu klik import. Maka akan muncul seperti Gambar 4.21. Pastikan tercentang semua lalu klik “Import”.



Gambar 4.21 Tampilan Import Package Integrasi Google Cardboard ke Unity

Langkah berikutnya adalah memasang kamera realitas virtual yang akan menggantikan kamera utama pada Unity. Langkah pertama adalah menghapus “*Main Camera*” pada scene Unity, lalu pasang kamera realitas virtual milik google dalam folder “*GoogleVR*”>”*Legacy*”>”*Prefabs*”>”*GvrMain*” pada scene Unity. Hasil dari penambahan kamera realitas virtual milik Google dapat dilihat pada Gambar 4.22.

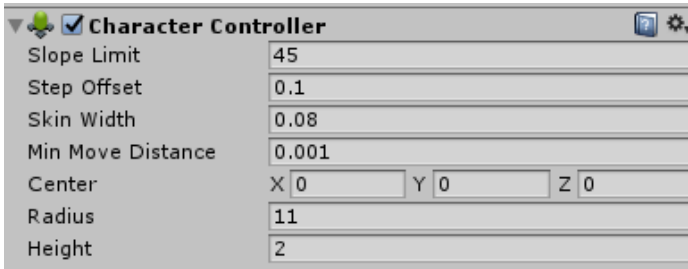


Gambar 4.22 Hasil Penambahan Kamera Realitas Virtual

4.4.2 Implementasi Kontrol Pengguna

Kontrol pengguna dalam permainan ini menggunakan kontroller nirkabel yang sudah terintegrasi dengan input control Unity dan memanfaatkan komponen Unity yaitu *Character Controller*. Hal yang perlu dilakukan adalah penambahan komponen pada kamera realitas virtual dengan cara klik “*Add Components*” pada inspector, lalu ketikkan “*Character Controller*” lalu tekan enter. Lalu berikan pengaturan sesuai

dengan keinginan. Hasil dari pengaturan dan penambahan komponen *Character Controller* terlihat seperti Gambar 4.23.



Gambar 4.23 Hasil Penambahan Komponen *Character Controller*

Selanjutnya agar dapat digerakkan menggunakan controller nirkabel, diperlukan sebuah *script* untuk membuat objek kamera virtual bergerak atau terbang. *Script* yang dibuat yaitu “*Fly.cs*”. *Script* tersebut akan diletakkan pada objek kamera sehingga dapat bergerak sesuai dengan input controller nirkabel. Fungsi untuk bergerak atau terbang dapat dilihat pada Kode Sumber 4.1.

```
public void Move(int direction, float
pushSpeed)
{
    if(direction == 0) //Vertical
    {
        moveDirection.x = 0;
        moveDirection.z = 1;
        cameraDirection =
gvrHead.transform.localPosition;
        if (cameraDirection.y > 0.01)
        {
            moveDirection.y = 1;
        }
        else if (cameraDirection.y <
-0.01)
```



```

        {
            moveDirection.y = -1;
        }
        else
        {
            moveDirection.y = 0;
        }
    }
    else if(direction == 1)
//Horizontal
    {
        moveDirection.x = 1;
        moveDirection.y = 0;
        moveDirection.z = 0;
        cameraDirection =
gvrHead.transform.localPosition;
    }
    moveSpeed = pushSpeed * speed;
    forward = vrHead.TransformDirection
(moveDirection);
    controller.Move (forward *
moveSpeed);
}

```

Kode Sumber 4.1 Fungsi Bergerak atau Terbang

4.4.3 Implementasi *Collider* Pengguna

Collider adalah komponen yang mensimulasikan adanya tabrakan antara objek satu dengan objek lainnya. Komponen ini diperlukan objek kamera virtual untuk bisa bertabrakan dengan objek lain dan dapat menghasilkan interaksi apabila terjadi tabrakan. Komponen *collider* yang digunakan adalah *Sphere Collider* yaitu *collider* yang berbentuk bulat yang mengelilingi objek kamera virtual. Cara menambahkan komponen ini adalah klik “*Add Component*” pada inspector, lalu ketikkan “*Sphere Collider*” lalu tekan enter. Komponen ini biasanya sudah terbentuk dengan ukuran sama dengan objek yang diberi komponen. Pastikan “*Is Trigger*” tidak tercentang agar objek

tidak tembus apabila terjadi tabrakan. Hasil dari penambahan komponen *Sphere Collider* dapat dilihat pada Gambar 4.24.



Gambar 4.24 Hasil Penambahan *Sphere Collider*

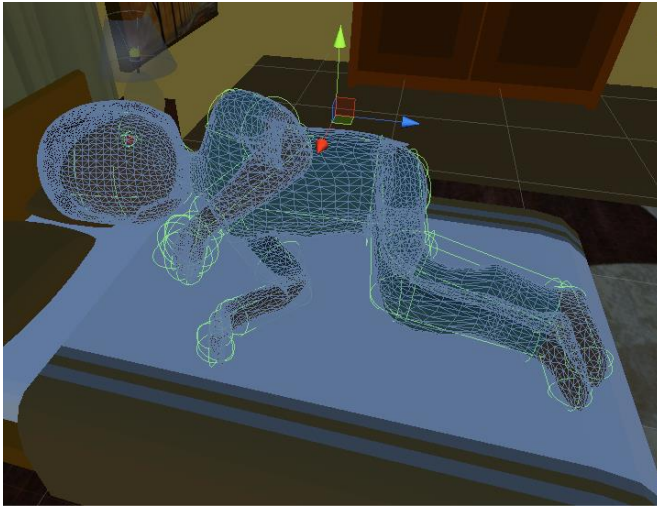
4.5 Implementasi Interaksi Objek Manusia

Pada tahap ini objek manusia diberikan interaksi antara objek manusia dengan karakter pengguna. Interaksi inilah yang menjadi elemen penting dari alur permainan yang dibuat. Pada awal tahap ini objek manusia diberikan komponen *collider* untuk setiap anggota badan agar dapat bersentuhan dengan karakter pengguna. Baik dari kepala, leher, badan, tangan, sampai ujung kaki. Selanjutnya objek manusia diberikan area untuk tempat menghisap darah, yang ditempatkan sesuai dengan skenario level yang dibuat sebelumnya. Setelah itu diberikan area pandangan manusia agar objek manusia dapat melihat karakter pengguna, dan terakhir diberikan alur animasi ketika terjadi interaksi dengan karakter pengguna. Untuk penjelasan lebih lanjut akan dijelaskan pada subbab berikut.

4.5.1 Implementasi *Collider* Objek Manusia

Pada tahap ini, setiap anggota tubuh objek manusia diberikan komponen *collider* agar dapat berinteraksi dengan karakter pengguna. Komponen *collider* yang digunakan adalah *capsule collider*. *Capsule collider* sama seperti *Sphere Collider* tetapi memiliki atribut ketinggian. Cara menambahkan *collider* sama seperti penambahan *collider* pada objek karakter

pengguna. Hasil dari penambahan komponen *collider* pada seluruh anggota tubuh objek manusia dapat dilihat pada Gambar 4.25.



Gambar 4.25 Hasil Penambahan *Collider* pada Seluruh Anggota tubuh Objek Manusia

4.5.2 Implementasi Area Tempat Menghisap Darah

Setelah penambahan *collider* pada objek manusia, tahap selanjutnya adalah penambahan area untuk menghisap darah objek manusia. untuk objek manusia ini, area untuk menghisap darah berada di pipi bagian kiri objek manusia. untuk menambahkan area, pertama yaitu memilih bagian wajah dari objek manusia lalu ditambahkan *GameObject* baru dengan cara klik menu “*Objects*”> “*Create Empty*”. Selanjutnya *GameObject* yang dibuat diberi komponen *Capsule Collider* dengan atribut “*Is Trigger*” dicentang agar dapat ditembus oleh objek karakter pengguna namun tetap terjadi interaksi apabila memasuki *collider*. Selanjutnya diberi *script* yaitu “*Hitzone.cs*” agar ketika karakter pengguna berada di area tersebut, karakter pengguna

akan menunjukkan notifikasi untuk memulai menghisap darah, dan ketika karakter pengguna keluar dari area, notifikasi akan menghilang. Fungsi untuk interaksi ketika karakter memasuki area dan keluar dari area terlihat pada Kode Sumber 4.2.

```
void OnTriggerEnter(Collider other)
{
    if(other.transform.tag == "Player")
    {
        Fly.onHitZone = true;
        Fly.hitZoneParent =
transform;
        Fly.notifChange = false;
        onHitzone = true;
    }
}

void OnTriggerExit(Collider other)
{
    if (other.transform.tag ==
"Player")
    {
        Fly.onHitZone = false;
        Fly.notifChange = false;
        onHitzone = false;
        anim.SetBool(animParameter,
false);
    }
}
```

Kode Sumber 4.2 Fungsi Interaksi Memasuki dan Keluar dari Area Menghisap Darah

Selanjutnya diberikan icon *sprite* berupa target merah untuk menunjukkan daerah yang dapat dihisap darahnya. Cara penambahan icon yaitu icon target merah dimasukkan ke dalam folder “Assets”, lalu klik icon yang dimasukkan tadi lalu klik “Texture Type” pada Inspector, lalu ganti dengan “Sprite (2D and

UI”, lalu masukkan icon ke dalam *scene* Unity dan letakkan ke dalam bagian wajah manusia di dalam *collider* area menghisap darah. Hasil dari tahap ini dapat terlihat pada Gambar 4.26.



Gambar 4.26 Hasil Penambahan Area Menghisap Darah pada Daerah Pipi Kiri Objek Manusia

4.5.3 Implementasi Area Pandangan Objek Manusia

Pada tahap ini objek manusia diberikan area pandangan di sekitar wajah. Area pandangan ini akan menambah kesadaran objek manusia apabila karakter manusia memasuki area tersebut dan akan berusaha mengusir karakter pengguna apabila pengguna terlalu lama di area. area ini diberi komponen *Sphere Collider* dengan atribut *Is Trigger* dicentang. Lalu agar objek manusia dapat menambah kesadaran apabila karakter pengguna memasuki area pandangan, diperlukan *script* yang bernama “*Shoozone.cs*”.

fungsi untuk interaksi karakter pengguna memasuki, diam, dan keluar dari area terlihat pada Kode Sumber 4.3.

```

void OnTriggerEnter(Collider other)
{
    if (other.transform.tag == "Player")
    {
        alertCountingTime = alertTime;
        onShooZone = true;
        AlertManager.IncreaseAlert(increaseAlertEnter)
    ;
    }
}

void OnTriggerStay(Collider other)
{
    if (other.transform.tag == "Player")
    {
        countingTime += Time.deltaTime;
        if (countingTime >= shooTime)
        {
            alertCountingTime -=
Time.deltaTime;
            if (alertCountingTime <= 1.0f)
            {
                AlertManager.IncreaseAlert(increaseAlertStay);
                alertCountingTime =
alertTime;
            }
            anim.SetBool("onShoozone", true);
        }
    }
}

void OnTriggerExit(Collider other)
{
    if (other.transform.tag == "Player")
    {
        onShooZone = false;
        alertCountingTime = 0;
        countingTime = 0;
        anim.SetBool("onShoozone", false);
    }
}

```

```

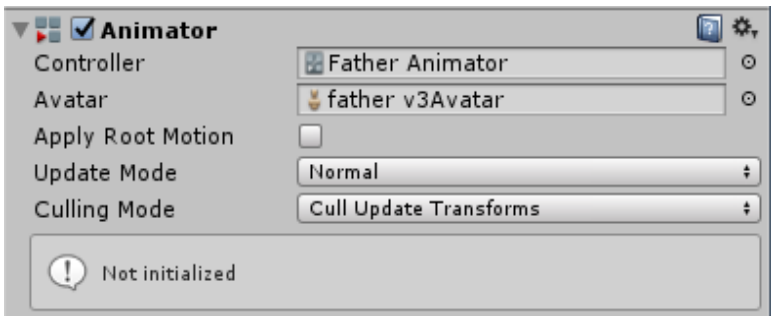
    }
}

```

Kode Sumber 4.3 Fungsi Interaksi Memasuki, Diam, dan Keluar dari Area Pandangan Objek Manusia

4.5.4 Implementasi Alur Animasi Objek Manusia

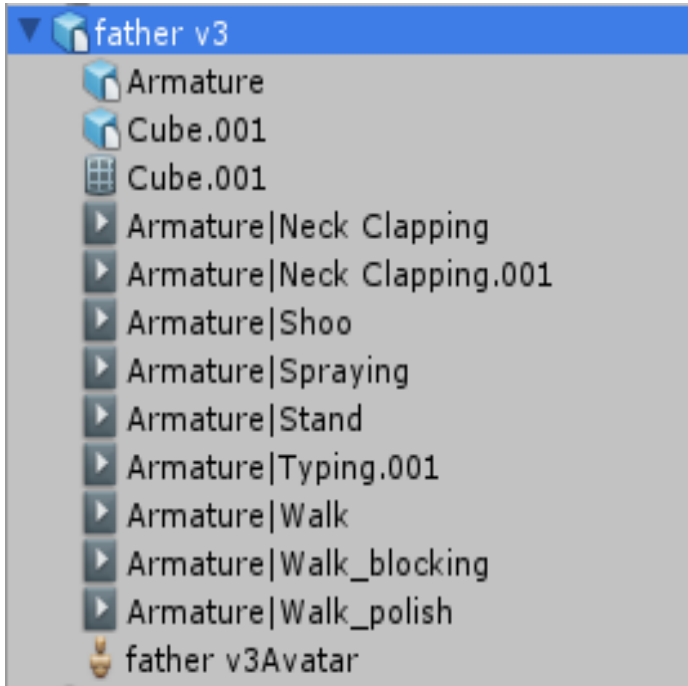
Pada tahap ini objek manusia diberi komponen *Animator* untuk membuat alur animasi yang akan berjalan ketika permainan dijalankan. Setiap objek manusia memiliki alur animasi yang berbeda-beda dan *Animator* yang berbeda pula. Untuk menambahkan atribut *Animator* untuk objek manusia pertama-pertama membuat file *Animator* terlebih dahulu yaitu klik kanan pada folder “Assets” di tab *Project*, lalu pilih “Create” > “Animator Controller”. Selanjutnya tambahkan komponen *Animator* pada objek manusia dengan cara drag animator yang dibuat ke dalam *inspector* object manusia. Hasil dari penambahan *Animator* pada objek manusia dapat dilihat pada Gambar 4.27.



Gambar 4.27 Hasil Penambahan Animator pada Objek Manusia

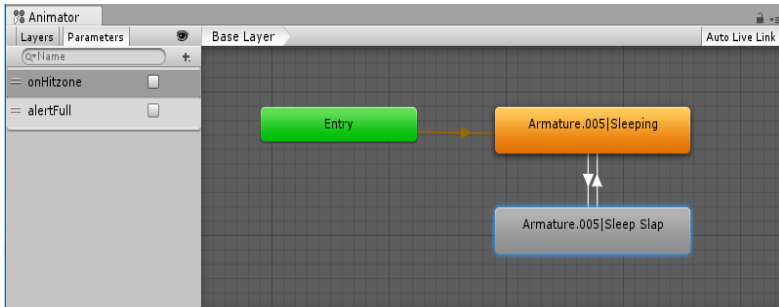
Setelah *Animator* ditambahkan pada objek manusia, masukkan animasi objek manusia tersebut yang dibuat di aplikasi

Blender ke dalam tab *Animation*. Cara untuk melakukannya yaitu buka folder dimana letak objek manusia itu disimpan di dalam folder “Assets” Unity, lalu klik panah disamping nama objek manusia, maka akan terlihat isi dari objek manusia termasuk animasinya seperti yang terlihat pada Gambar 4.28.



Gambar 4.28 Isi dari Objek Manusia termasuk Animasinya

Setelah itu masukkan file animasi ke dalam tab *Animation*. Lalu diatur animasi mana yang menjadi animasi utama ketika permainan dijalankan. Buat transisi dari animasi satu ke animasi yang lain dengan cara klik kanan pada animasi lalu klik “*Make Transition*”. Tambahkan variabel untuk perpindahan animasi satu ke animasi yang lain. Hasil dari alur animasi yang dibuat terlihat seperti pada Gambar 4.29.



Gambar 4.29 Hasil Alur Animasi pada Objek Manusia

Dari alur animasi yang dibuat, objek manusia akan melakukan aktifitas tidur sebagai aktifitas utamanya, lalu manusia akan memukul daerah pipi apabila variabel “*OnHitzone*” berisi *True*, dimana manusia akan memukul karakter pengguna apabila berada di daerah menhisap darah.

4.6 Implementasi Interaksi Objek Tak Bergerak

Pada tahap ini Objek yang tidak bergerak ditambahkan komponen *collider* dan penambahan *script* untuk bersembunyi dari racun dan mengurangi kesadaran objek manusia pada beberapa objek tak bergerak. *Collider* yang digunakan untuk objek tak bergerak disesuaikan dengan bentuk objek. Apabila bentuk objek terdiri dari kubus atau balok, maka *collider* yang digunakan adalah *Box Collider*. Apabila objek terbuat dari bola, atau tabung maka *collider* yang digunakan adalah *Capsule Collider*. *Collider* diberikan agar objek tak bergerak tidak dapat ditembus oleh objek karakter pengguna. Setelah diberikan *collider*, beberapa objek tak bergerak diberi *script* bernama “*Hidezone.cs*” yang bertujuan sebagai tempat bersembunyi karakter pengguna agar tidak terkena racun dan mengurangi kesadaran manusia apabila karakter memasuki area. Fungsi ketika karakter memasuki dan keluar dari area sembunyi terlihat pada Kode Sumber 4.4. Hasil dari penambahan *collider* dan *script* area bersembunyi terlihat seperti pada Gambar 4.30.

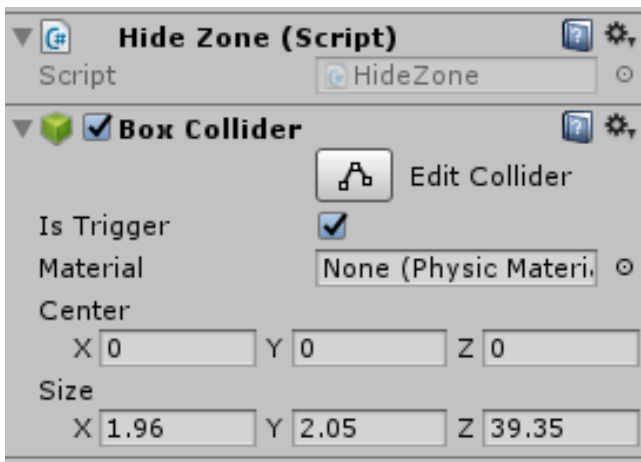
```

void OnTriggerEnter(Collider other)
{
    if(other.tag == "Player")
    {
        AlertManager.inHideZone = true;
    }
}

void OnTriggerExit(Collider other)
{
    if (other.tag == "Player")
    {
        AlertManager.inHideZone = false;
    }
}

```

Kode Sumber 4.4 Fungsi Interaksi Memasuki, dan Keluar dari Area Sembunyi



Gambar 4.30 Hasil Penambahan *Collider* dan *Script* pada Objek Tak Bergerak

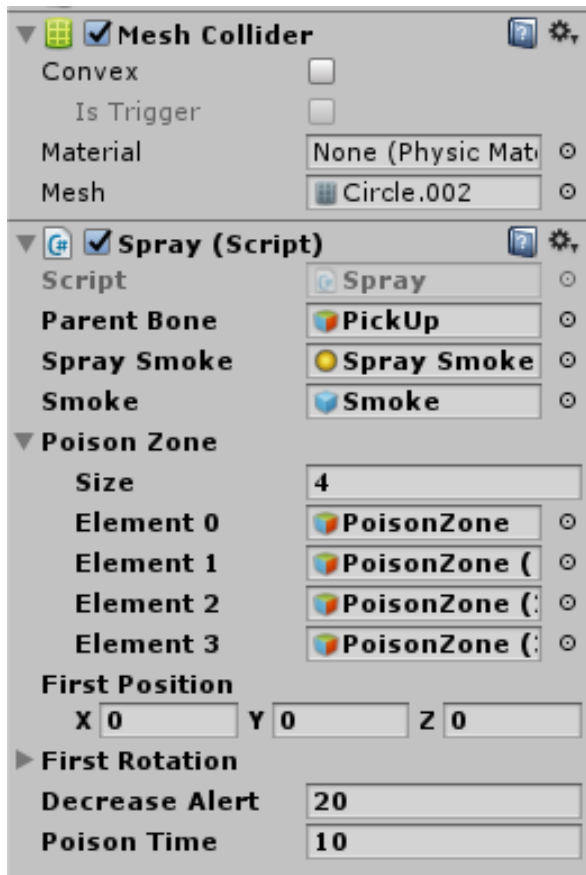
4.7 Implementasi Interaksi Objek Bergerak

Pada tahap ini dilakukan penambahan *collider* dan *script* untuk menggerakkan objek bergerak. Untuk objek bergerak yang dibuat dan digunakan pertama kali dalam permainan adalah obat nyamuk atau *spray*. Untuk penambahan *collider* sama seperti objek tak bergerak, yaitu menambahkan tipe *collider* yang cocok dengan bentuk objek bergerak. Untuk objek obat nyamuk, *collider* yang digunakan adalah *mesh collider*, dimana *collider* ini berbentuk menyesuaikan bentuk objek. Setelah penambahan komponen *collider*, objek bergerak diberi *script* untuk bisa digerakkan oleh manusia. *Script* tersebut diberi nama “*Spray.cs*”. Fungsi untuk menggerakkan objek bergerak terlihat pada Kode Sumber 4.5. Hasil dari penambahan *collider* dan *Script* penggerak pada objek bergerak terlihat pada Gambar 4.31.

```
public void PickUpSpray()
{
    transform.parent = parentBone.transform;
    transform.position =
parentBone.transform.position;
    transform.rotation =
parentBone.transform.rotation;
}

public void PutDownSpray()
{
    transform.parent = firstParent;
    transform.position = firstPosition;
    transform.rotation = firstRotation;
    AlertManager.DecreaseAlert(decreaseAlert);
}
```

Kode Sumber 4.5 Fungsi Penggerak Objek Bergerak



Gambar 4.31 Hasil Penambahan *Collider* dan *Script* pada Objek Bergerak

4.8 Implementasi Alur Permainan

Pada tahap ini akan dijelaskan bagaimana implementasi pada alur permainan realitas virtual. Alur permainan ini terdiri dari beberapa komponen yang dapat mempengaruhi alur permainan. Tahap ini akan dijelaskan lebih lanjut pada subbab-subbab berikut.

4.8.1 Implementasi Waktu Permainan

Pada tahap ini akan dijelaskan mengenai waktu permainan. Waktu permainan adalah waktu yang diberikan saat pengguna memainkan satu level. Waktu yang diberikan dalam satu level sebanyak 200 detik. Apabila waktu telah habis, maka permainan berakhir dan pengguna harus mengulangi level dari awal dan waktu akan di-*reset*. Implementasi tahap ini menggunakan *script* yang bernama “*TimeManager*”. Fungsi untuk *script* menjalankan dan me-*reset* waktu permainan dapat dilihat pada Kode Sumber 4.6.

```
void Start()
{
    countingTime = startingTime;
    theText = GetComponent<TextMesh>();
}

// Update is called once per frame
void Update()
{
    countingTime -= Time.deltaTime;
}

public void ResetTime()
{
    countingTime = startingTime;
}
```

Kode Sumber 4.6 Fungsi Menjalankan dan me-*reset* Waktu Permainan

4.8.2 Implementasi Nyawa Objek Pengguna

Pada tahap ini dijelaskan tentang implementasi nyawa objek karakter pengguna dalam permainan, dalam satu level, karakter pengguna diberikan 3 nyawa. Nyawa tersebut dapat berkurang sejumlah setengah atau satu. Apabila nyawa habis,

maka permainan selesai dan pengguna harus mengulangi level dari awal. Implementasi ini menggunakan *script* yang bernama “*HealthManager.cs*”. Fungsi untuk menjalankan dan mengurangi nyawa karakter pengguna dapat dilihat pada Kode Sumber 4.7.

```
void Start()
{
    initialHealth = 3;
    health = initialHealth;
}

public static void DecreaseHealth(float
healthToDecrease)
{
    health -= healthToDecrease;
    countingTime = staticHurtTime;
    healthChange = false;
}
```

Kode Sumber 4.7 Fungsi Menjalankan dan Mengurangi Nyawa Pengguna

4.8.3 Implementasi Melukai Objek Pengguna

Objek karakter pengguna dapat dilukai oleh tangan objek manusia apabila manusia merasa terganggu dan berusaha memukul karakter pengguna dan karakter pengguna mengenai pukulan objek manusia. Apabila karakter pengguna terkena pukulan manusia maka nyawa karakter pengguna berkurang satu. Implementasi ini menggunakan *script* yang bernama “*HurtPlayer.cs*”. Fungsi untuk melukai objek pengguna dapat dilihat pada Kode Sumber 4.8.

```
void OnTriggerEnter (Collider other)
{
    if (other.transform.tag == "Player")
```

```

    {
        if (Fly.onHitZone)
            HealthManager.isDead =
true;
        if (shooZone.onShooZone &&
!fly.playerHurt)

            HealthManager.DecreaseHealth(1);
    }
}

```

Kode Sumber 4.8 Fungsi Melukai Objek Pengguna

4.8.4 Implementasi Kesadaran Objek Manusia

Pada tahap ini akan dijelaskan implementasi kesadaran objek manusia apabila karakter pengguna berada di sekitar objek manusia. Kesadaran objek manusia dapat bertambah apabila karakter pengguna memasuki area pandangan objek manusia, diam selama beberapa detik di area pandangan, atau melakukan kesalahan pada *minigame*. Kesadaran objek manusia bisa berkurang apabila manusia telah menggunakan obat nyamuk atau karakter pengguna bersembunyi di area bersembunyi selama beberapa waktu. Implementasi ini menggunakan *script* yang bernama "AlertManager.cs". Fungsi untuk menjalankan dan mengurangi kesadaran objek manusia dapat dilihat pada Kode Sumber 4.9.

```

void Start ()
{
    initialAlert = 0;
    alert = initialAlert;
}

void Update ()
{
    alertText.text = alert.ToString();
}

```

```

public static void IncreaseAlert(int
pointToIncrease)
{
    alert += pointToIncrease;
}

public static void DecreaseAlert(int
pointToDecrease)
{
    alert -= pointToDecrease;
}

```

Kode Sumber 4.9 Fungsi Menjalankan dan Mengurangi Kesadaran Objek Manusia

4.8.5 Implementasi Darah yang Terhisap dari Objek Manusia

Pada tahap ini akan dijelaskan implementasi darah yang terhisap dari objek manusia. Darah didapatkan dengan cara menghisap darah pada daerah menghisap pada suatu bagian tubuh manusia. Apabila darah mencapai nilai 100, maka permainan selesai dan pengguna dapat melanjutkan ke level selanjutnya. Implementasi ini menggunakan *script* yang bernama “*BloodManager.cs*”. Fungsi untuk menjalankan dan menambah darah dapat dilihat pada Kode Sumber 4.10.

```

void Start ()
{
    initalBlood = 0;
    blood = initalBlood;
    bloodText =
GetComponent<TextMesh>();
    bloodText.text = blood.ToString();
}

void Update ()

```



```

{
    bloodText.text = blood.ToString();
}

public static void IncreaseBlood(int
pointToIncrease)
{
    blood += pointToIncrease;
}

```

Kode Sumber 4.10 Fungsi Menjalankan dan Menambah Darah

4.8.6 Implementasi Racun

Pada tahap ini akan dijelaskan implementasi dari racun. Racun dapat mengurangi nyawa karakter pengguna sebanyak setengah bar apabila karakter pengguna berada di area beracun selama beberapa waktu. Racun dapat dikeluarkan apabila kesadaran objek manusia mencapai 100. Racun akan menghilang selama beberapa detik. Cara untuk menghindari racun adalah bersembunyi di area bersembunyi. Implementasi ini menggunakan beberapa *script*, yaitu “*Human.cs*” untuk mengambil objek obat nyamuk dan menjalankan animasi menyemprotkan obat nyamuk, “*Spray.cs*” untuk bergerak sesuai dengan animasi objek manusia menyemprotkan racun, mengeluarkan racun, dan mengaktifkan racun, “*PoisonZone.cs*” untuk interaksi masuk, atau keluar dari zona racun, dan “*HealthManager.cs*” untuk membuat karakter pengguna terkena racun dan mulai mengurangi nyawa karakter. Fungsi pada *script* “*Human.cs*” dapat dilihat pada Kode Sumber 4.11. *Script* “*Spray.cs*” pada Kode Sumber 4.12. “*PoisonZone.cs*” pada Kode Sumber 4.13, dan “*HealthManager.cs*” pada Kode Sumber 4.14.

```

void Update ()
{
    if (AlertManager.alertOver100)

```

```

        {
            anim.SetBool ("alertFull", true);
        }
        else if(!AlertManager.alertOver100)
        {
            anim.SetBool ("alertFull",
false);
        }
    }

    public void PickUpSpray()
    {
        spray.PickUpSpray();
    }

    public void Spraying()
    {
        spray.Spraying();
    }

    public void PutDownSpray()
    {
        spray.PutDownSpray();
    }
}

```

Kode Sumber 4.11 Fungsi pada script *Human.cs*

```

public void Spraying()
{
    smoke.SetActive(true);
    smoke.GetComponent<ParticleSystem>().Play(
);
    StartCoroutine(PoisonStart(poisonTime));
}

public IEnumerator PoisonStart(float time)
{
    HealthManager.poisonActive = true;
    //poisonIcon.SetActive(true);
    for(int i=0;i<poisonZone.Length;i++)
    {

```

```

        poisonZone[i].SetActive(true);
    }
    yield return new WaitForSeconds(time);
    for (int i = 0; i < poisonZone.Length;
i++)
    {
        poisonZone[i].SetActive(false);
    }
    HealthManager.poisonActive = false;
    //poisonIcon.SetActive(false);
}

```

Kode Sumber 4.12 Fungsi pada *script Spray.cs*

```

void OnTriggerStay(Collider other)
{
    if (other.tag == "Player")
    {
        HealthManager.inPoisonZone =
true;
    }
}

void OnTriggerExit(Collider other)
{
    if (other.tag == "Player")
    {
        HealthManager.inPoisonZone =
false;
    }
}

```

Kode Sumber 4.13 Fungsi pada *script PoisonZone.cs*

```

void Update ()
{

```

```

        if (inPoisonZone && !isPoisoned)
        {
            StartDecreaseHealth
(poisonHealthVal, poisonTime);
        }
        if (!inPoisonZone && isPoisoned)
        {
            StopDecreaseHealth
(poisonHealthVal, poisonTime);
        }
    }

```

Kode Sumber 4.14 Fungsi pada *script HealthManager.cs*

4.8.7 Implementasi *Minigame*

Pada tahap ini akan dijelaskan implementasi *minigame*. *Minigame* akan dijalankan apabila karakter pengguna berada di area menghisap darah dan memulai menghisap darah objek manusia. Untuk memulai *minigame*, pengguna cukup menekan tombol “A” atau “X”. *Minigame* pada permainan ini berupa permainan kecil dengan cara menekan tombol arah yang muncul pada layar kamera realitas virtual pengguna. Apabila tombol yang muncul pada *minigame* ditekan sesuai arahnya, maka pengguna akan mendapatkan darah dari objek manusia, tetapi apabila menekan tombol yang salah, maka darah tidak akan bertambah tetapi kesadaran manusia akan bertambah. Dan apabila pengguna masih dalam permainan *minigame* dan kesadaran objek manusia mencapai 100, maka objek manusia memukuli karakter pengguna dan permainan akan dinyatakan kalah dan pengguna harus mengulang level dari awal. Implementasi ini menggunakan beberapa *script* yaitu “*RhythmMiniGame.cs*” untuk memulai dan menghentikan *minigame*, “*RhythmGameButton.cs*” untuk input tombol pada *minigame*, “*Hitzone.cs*” untuk menjalankan animasi memukul karakter apabila kesadaran mencapai 100, dan “*Fly.cs*” untuk input tombol saat karakter pengguna ingin memulai menghisap darah dan mulai memainkan *minigame*. Fungsi pada *script* “*RhythmMiniGame.cs*” dapat dilihat pada Kode Sumber

4.15. Script “*RhythmGameButton.cs*” pada Kode Sumber 4.16. “*Hitzone.cs*” pada Kode Sumber 4.17, dan “*Fly.cs*” pada Kode Sumber 4.18.

```
public void StartMiniGame()
{
    miniGameStart = true;
    miniGameScreen.SetActive (true);
    StartCoroutine(MiniGameStart());
}

public void StopMiniGame()
{
    miniGameStart = false;
    StopCoroutine(MiniGameStart());
    miniGameScreen.SetActive (false);
    Debug.Log("Mini Game Stopped");
}

public IEnumerator MiniGameStart()
{
    Debug.Log("Minigame start");
    int x = Random.Range(0, 9);
    float time = 1f;
    yield return new WaitForSeconds(time);
    while(miniGameStart)
    {
        if (miniGameStart)
        {
            int y = Random.Range(0, 4);
            Debug.Log(y);
            if (y == 4)
            {
                y = Random.Range(0, 3);
            }
            button[y].SetActive(true);
            yield return new
WaitForSeconds(time);
            button[y].SetActive(false);
        }
    }
}
```

```

        button[y].GetComponent<RhythmGameButton>().
        .enabled = true;

        button[y].GetComponent<SpriteRenderer>().c
        olor = initialColor;
        yield return new
        WaitForSeconds(time);
        }
        else
        yield return 0;
    }
}

public void ButtonSuccess()
{
    AlertManager.IncreaseAlert(increaseAlertSu
    ccess);
    BloodManager.IncreaseBlood(increaseBlood);
}

public void ButtonFailed()
{
    AlertManager.IncreaseAlert(increaseAlertFa
    iled);
}

```

Kode Sumber 4.15 Fungsi pada script *RhythmMiniGame.cs*

```

void Update ()
{
    if(Input.GetAxis("Vertical") > 0.1)
    {
        if(whatButton == 1)
        {
            sprite.color =
            Color.green;

            rhythmMiniGame.ButtonSuccess();
        }
    }
}

```

```

        }
        else
        {
            sprite.color = Color.red;

            rhythmMiniGame.ButtonFailed();
        }
        this.enabled = false;
    }

    if(Input.GetAxis("Vertical") < -0.1)
    {
        if (whatButton == -1)
        {
            sprite.color =
Color.green;

            rhythmMiniGame.ButtonSuccess();
        }
        else
        {
            sprite.color = Color.red;

            rhythmMiniGame.ButtonFailed();
        }
        this.enabled = false;
    }

    if (Input.GetAxis("Horizontal") > 0.1)
    {
        if (whatButton == 2)
        {
            sprite.color =
Color.green;

            rhythmMiniGame.ButtonSuccess();
        }
        else
        {
            sprite.color = Color.red;

            rhythmMiniGame.ButtonFailed();
        }
        this.enabled = false;
    }

```

```

    }

    if (Input.GetAxis("Horizontal") < -0.1)
    {
        if (whatButton == -2)
        {
            sprite.color =
Color.green;

            rhythmMiniGame.ButtonSuccess();
        }
        else
        {
            sprite.color = Color.red;

            rhythmMiniGame.ButtonFailed();
        }
        this.enabled = false;
    }
}

```

Kode Sumber 4.16 Fungsi pada script *RhythmGameButton.cs*

```

void Update ()
{
    if(HealthManager.isDead)
    {
        anim.SetBool(animParameter,
false);
    }
    if(Fly.onHitZone &&
AlertManager.GetAlert() >= 100 &&
!HealthManager.isDead && onHitzone)
    {
        anim.SetBool(animParameter, true);
    }
}

```

Kode Sumber 4.17 Fungsi pada script *Hitzone.cs*


```

void Update ()
{
    if (Input.GetButtonDown ("Fire1") &&
onHitZone)
    {
        if (!rhythmMiniGame.miniGameStart)
        {

            transform.parent = hitZoneParent;

            flyAnim.SetBool ("isSucking", true);

            rhythmMiniGame.StartMiniGame ();
            notifChange = false;

            ChangeNotification (false, "");
        }
        else if (rhythmMiniGame.miniGameStart)
        {

            transform.parent = firstParent;

            //rhythmMiniGame.miniGameStart = false;

            flyAnim.SetBool ("isSucking", false);
            notifChange = false;

            rhythmMiniGame.StopMiniGame ();
        }
    }
    if (onHitZone && !notifChange)
    {
        ChangeNotification (true, "Press X/A
Button to Start Sucking Blood");
    }
    else if (!onHitZone && !notifChange)
    {
        if (rhythmMiniGame.miniGameStart)
        {

            transform.parent = firstParent;

            rhythmMiniGame.StopMiniGame ();
        }
    }
}

```

```
ChangeNotification (false, "");
}
```

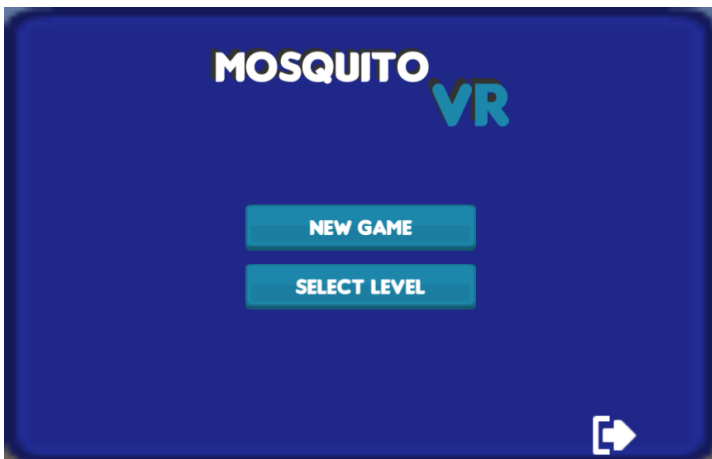
Kode Sumber 4.18 Fungsi pada *script Fly.cs*

4.9 Implementasi Antarmuka

Pada tahap ini akan dijelaskan antarmuka pada permainan realitas virtual yang dibangun oleh penulis. Antarmuka yang diimplementasikan dalam permainan ini adalah menu utama, layar tunggu (*loading screen*), antarmuka dalam permainan (*ingame*), dan antarmuka *minigame*. Tahapan ini akan dijelaskan lebih lanjut pada subbab-subbab berikut.

4.9.1 Implementasi Antarmuka Menu Utama

Pada tahap ini akan dijelaskan tentang implementasi antarmuka menu utama. Antarmuka ini terdiri dari tiga tombol yaitu “New Game”, “Load Game”, dan “Exit”. Tampilan dari antarmuka ini ditunjukkan pada gambar 4.32.



Gambar 4.32 Tampilan Antarmuka Menu Utama

Tombol “*New Game*” digunakan untuk masuk ke level pertama dalam permainan. Sebelum memasuki level permainan, tombol ini akan mengarah ke antarmuka layar tunggu (*Loading Screen*) untuk memproses *load* dari level yang akan dimainkan. *Script* untuk tombol “*New Game*” dapat dilihat pada Kode Sumber 4.19.

```
public void NewGame ()
{
    LoadingScreenManager.LoadScene (1);
    PlayerPrefs.SetInt ("atLevel", 1);
}
```

Kode Sumber 4.19 Fungsi Tombol “*New Game*”

Tombol “*Select Level*” digunakan untuk memilih level yang ingin dimainkan. Apabila ditekan, maka akan muncul beberapa tombol lagi untuk memilih level yang diinginkan. Sama seperti tombol “*New Game*”, tombol level akan mengarah ke antarmuka layar tunggu untuk memproses level yang akan dimainkan. Potongan kode untuk tombol “*Select Level*” dan pemilihan Level dapat dilihat pada Kode Sumber 4.20.

```
public void SelectLevel ()
{
    anim.SetBool ("selectLevel", true);
}

public void Back ()
{
    anim.SetBool ("selectLevel",
false);
}

public void Level1 ()
{
    LoadingScreenManager.LoadScene (1);
}
```

```

        PlayerPrefs.SetInt ("atLevel", 1);
    }

    public void Level2()
    {
        LoadingScreenManager.LoadScene(2);
        PlayerPrefs.SetInt ("atLevel", 2);
    }

    public void Level3()
    {
        LoadingScreenManager.LoadScene(3);
        PlayerPrefs.SetInt ("atLevel", 3);
    }

```

Kode Sumber 4.20 Fungsi Tombol “*Select Level*” dan Pemilihan Level

Tombol “*Exit*” yang dilambangkan icon pintu digunakan untuk keluar dari permainan. Potongan kode untuk tombol “*Exit*” dapat dilihat pada Kode Sumber 4.21.

```

public void Quit()
{
    Application.Quit ();
}

```

Kode Sumber 4.21 Fungsi Tombol “*Exit*”

4.9.2 Implementasi Antarmuka Layar Tunggu

Pada tahap ini akan dijelaskan implementasi antarmuka layar tunggu. Antarmuka ini akan muncul ketika pengguna memilih level dan sistem sedang menyiapkan level yang akan dimainkan. Tampilan antarmuka layar tunggu dapat dilihat pada Gambar 4.33.



Gambar 4.33 Tampilan Antarmuka Layar Tunggu

Saat sistem mempersiapkan level yang akan dimainkan, antarmuka layar tunggu akan menampilkan fakta menarik seputar hewan nyamuk. Fakta yang ditampilkan hanya satu dalam satu kali masuk ke antarmuka layar tunggu. Tampilan layar tunggu ketika menampilkan fakta menarik dapat dilihat pada gambar sebelumnya, untuk potongan kode dari menampilkan fakta dapat dilihat pada Kode Sumber 4.22.

```
public void ShowFact ()
{
    didYouKnow.gameObject.SetActive
(true);
    funFactImage.gameObject.SetActive
(true);
    funFacts.gameObject.SetActive
(true);
}
```

Kode Sumber 4.22 Fungsi Menampilkan Fakta

Apabila level sudah dipersiapkan dan siap untuk dimainkan, antarmuka layar tunggu akan menampilkan perintah kepada pengguna untuk memasukkan *smartphone* ke dalam *google cardboard*. Sistem memberikan waktu tunggu selama 10 detik kepada pengguna untuk memasukkan ke dalam *google cardboard* sebelum permainan dimulai. Tampilan dari layar tunggu ketika level sudah dipersiapkan terlihat seperti pada gambar 4.32. Untuk fungsi menampilkan perintah untuk bersiap-siap memasukkan *smartphone* ke dalam *google cardboard* dapat dilihat pada Kode Sumber 4.23. Untuk proses *load* level yang akan dimainkan, dapat dilihat pada Kode Sumber 4.24.

```
void ShowCompletionVisuals () {
    loadingIcon.gameObject.SetActive (false);
    loadingDoneIcon.gameObject.SetActive (true)
};

    CountdownText.gameObject.SetActive
(true);
    CountdownTime.gameObject.SetActive
(true);

    loadingText.gameObject.SetActive (false);
    progressBar.gameObject.SetActive (false);
    progressBarHole.gameObject.SetActive
(false);

    didYouKnow.gameObject.SetActive (false);
    funFactImage.gameObject.SetActive
(false);
    funFacts.gameObject.SetActive (false);

    cardboard.gameObject.SetActive (true);
    preparation.gameObject.SetActive (true);
    instruction.gameObject.SetActive (true);
    CountdownTime.text =
waitOnLoadEnd.ToString ();
    //progressBar.fillAmount = 1f;
```

```

        //loadingText.text = "LOADING DONE";
    }

```

Kode Sumber 4.23 Fungsi Menampilkan Perintah Bersiap-siap Memasuki Level.

```

private IEnumerator LoadAsync(int
levelNum) {
    ShowLoadingVisuals();

    yield return null;

    FadeIn();
    StartOperation(levelNum);

    float lastProgress = 0f;

    // operation does not auto-activate
    scene, so it's stuck at 0.9
    while (DoneLoading() == false) {
        yield return null;

        if
(Mathf.Approximately(operation.progress,
lastProgress) == false) {
            progressBar.fillAmount
= operation.progress;
            lastProgress =
operation.progress;
        }
    }

    if (loadSceneMode ==
LoadSceneMode.Additive)
        audioListener.enabled =
false;

    ShowCompletionVisuals();
    loadingDone = true;
}

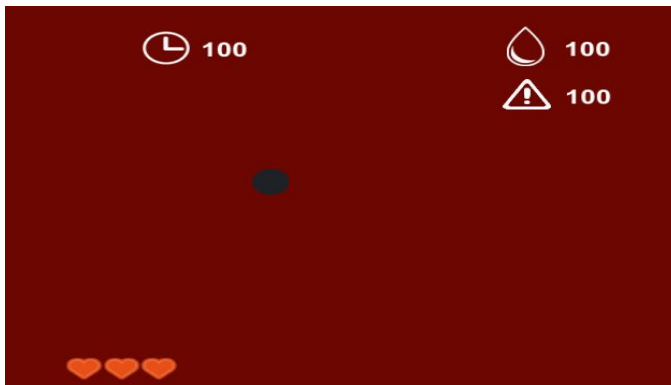
```

```
}
```

Kode Sumber 4.24 Fungsi Proses Sistem Me-load Level

4.9.3 Implementasi Antarmuka Karakter Pengguna

Pada tahap ini akan dijelaskan tentang antarmuka ketika permainan sudah dimulai. Antarmuka ini ditampilkan pada karakter Pengguna yang dimainkan oleh pengguna. Tampilan antarmuka karakter pengguna dapat dilihat pada Gambar 4.34.



Gambar 4.34 Tampilan Antarmuka Karakter Pengguna

Pada tampilan antarmuka ini, terdapat ikon dan angka yang menjelaskan tentang elemen permainan. Yaitu teks notifikasi, ikon nyawa pemain (*Health*), ikon sembunyi (*Hide*), ikon racun (*Poison*), Waktu Permainan (*Time*), Darah yang dihisap (*Blood*), kesadaran objek manusia (*Alert*). Ikon nyawa akan mengikuti jumlah nyawa yang dimiliki pengguna, lalu ikon sembunyi akan muncul ketika pengguna berada di tempat besembunyi, ikon racun akan muncul ketika objek manusia menyemprotkan racun, waktu permainan akan berubah menjadi merah apabila waktu kurang dari 30 detik, ikon kesadaran objek manusia akan berubah menjadi kuning apabila mencapai 50 dan

menjadi merah apabila mencapai 100. Implementasi dari antarmuka ini terdapat pada beberapa *script*, yaitu “*Fly.cs*” untuk teks notifikasi, “*HealthManager.cs*” untuk nyawa pemain dan ikon racun, “*AlertManager.cs*” untuk kesadaran objek manusia dan ikon sembunyi, “*BloodManager.cs*” untuk darah yang dihisap. Untuk *script* “*Fly.cs*” dapat dilihat pada Kode Sumber 4.24, *script* “*HealthManager.cs*” pada Kode Sumber 4.25, “*AlertManager.cs*” pada Kode Sumber 4.26, dan “*BloodManager.cs*” pada Kode Sumber 4.27.

```
public void ChangeNotification(bool flag,
string text)
{
    notification.text = text;
    notificationUI.SetActive (flag);
    notifChange = true;
}
```

Kode Sumber 4.25 Fungsi Menampilkan Notifikasi

```
void Update ()
{
    if (poisonActive && !poisonIconChange)
    {
        poisonIcon.SetActive (true);
        poisonIconChange = true;
    }
    else if (!poisonActive &&
poisonIconChange)
    {
        poisonIcon.SetActive (false);
        poisonIconChange = false;
    }

    if(inPoisonZone && !isPoisoned)
    {
        StartDecreaseHealth
    }
}
```

```

(poisonHealthVal, poisonTime);
    }
    if (!inPoisonZone && isPoisoned)
    {
        StopDecreaseHealth
(poisonHealthVal, poisonTime);
    }

    if (!healthChange)
    {
        ChangeHealthIcon (health);
    }
    else if (health <= 0.0f || isDead)
    {
        ChangeHealthIcon (0.0f);
    }
}

public void StartDecreaseHealth(float
pointToDecrease, float time)
{
    isPoisoned = true;
    StartCoroutine(DecreaseHealthPerSec(point
ToDecrease, time));
}

public void StopDecreaseHealth(float
pointToDecrease, float time)
{
    isPoisoned = false;
    StopCoroutine(DecreaseHealthPerSec(pointT
oDecrease, time));
}

public IEnumerator DecreaseHealthPerSec(float
pointToDecrease, float time)
{
    while (isPoisoned)
    {

```

```

        health -= pointToDecrease;
        healthChange = false;
        yield return new
WaitForSeconds(time);
    }
}
}

```

Kode Sumber 4.26 Fungsi Menampilkan Nyawa dan Ikon Racun

```

void Update ()
{
    alertText.text = alert.ToString();

    if (inHideZone && !isHiding)
    {
        shieldIcon.SetActive (true);
        StartDecreaseAlert (hideAlertVal,
hideTime);
    }

    else if (!inHideZone && isHiding)
    {
        shieldIcon.SetActive (false);
        StopDecreaseAlert (hideAlertVal,
hideTime);
    }

    if(alert >=100)
    {
        alert = 100;
        if (!alertOver100)
        {
            changeAlertIcon ();
        }
    }
}

```

```

        else if(alert < 0)
        {
            alert = 0;
        }

        else
        {
            if (alert > 50 && alert < 100 &&
!alertOver50)
            {
                changeAlertIcon ();
            }

            else if (alert < 50 && !alertOver0)
            {
                changeAlertIcon ();
            }
        }
    }

    public void StartDecreaseAlert(int
pointToDecrease,float time)
    {
        isHiding = true;
        StartCoroutine(DecreaseAlertPerSec(pointT
oDecrease,time));
    }

    public void StopDecreaseAlert(int
pointToDecrease, float time)
    {
        isHiding = false;
        StopCoroutine(DecreaseAlertPerSec(pointTo
Decrease, time));
    }

    public IEnumerator DecreaseAlertPerSec(int
pointToDecrease,float time)
    {
        while (isHiding)

```

```

{
    if(alert > 0)
        alert -= pointToDecrease;

    yield return new
WaitForSeconds(time);
}
}

```

Kode Sumber 4.27 Fungsi Menampilkan Kesadaran dan Ikon Sembunyi

```

void Update ()
{
    bloodText.text = blood.ToString();

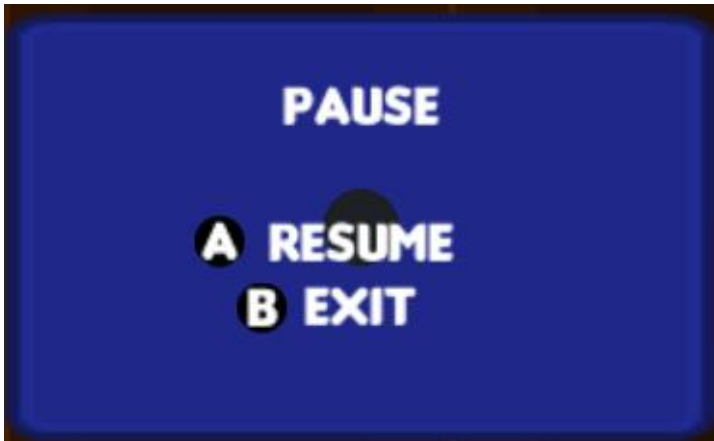
    if (blood >= 100)
    {
        blood = 100;
        bloodFull = true;
    }

    if (bloodFull && !isWin)
    {
        WinStage ();
    }
}

```

Kode Sumber 4.28 Fungsi Menampilkan Darah yang Dihisap

Selain antarmuka diatas, terdapat antarmuka ketika pengguna menekan tombol “*start*” pada kontroller. Antarmuka yang akan muncul adalah antarmuka *pause* atau menghentikan permainan sementara. Tampilan antarmuka ini dapat dilihat pada Gambar 4.35.



Gambar 4.35 Tampilan Antarmuka *Pause*

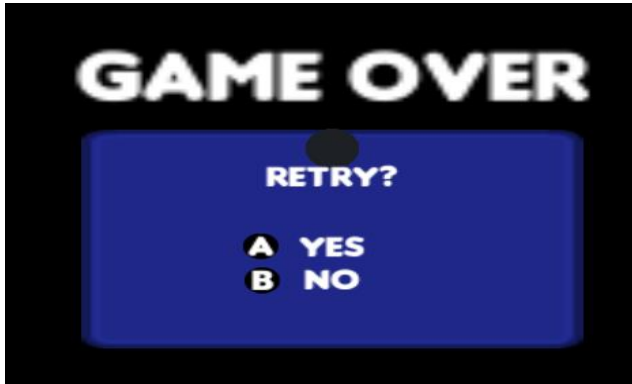
Pada antarmuka ini hanya ada 2 perintah yang dapat dijalankan dengan menekan tombol “A” atau “B” pada kontroller, yaitu melanjutkan permainan (*resume game*) atau keluar dari level (*quit game*), tombol “A” untuk melanjutkan permainan, sedangkan “B” untuk kembali ke menu utama. *Script* untuk antarmuka pause dapat dilihat pada kode sumber 4.29.

```
void Update ()
{
    if(Input.GetButtonDown("Fire1"))
    {
        player.Pause ();
    }

    if (Input.GetButtonDown ("Fire2"))
    {
        SceneManager.LoadScene ("Main
Menu");
    }
}
```

Kode Sumber 4.29 *Script* Antarmuka *Pause*

Apabila pengguna mengalami kekalahan seperti nyawa karakter pengguna habis, atau waktu habis, maka akan muncul antarmuka permainan kalah (*gameover*), yang ditampilkan seperti pada Gambar 4.36.



Gambar 4.36 Tampilan Antarmuka *Gameover*

Pada antarmuka ini hanya ada 2 perintah yang dapat dijalankan dengan menekan tombol “A” atau “B” pada kontrollor, yaitu mengulang permainan (*retry*) atau keluar dari level (*quit game*), tombol “A” untuk mengulang permainan, sedangkan “B” untuk kembali ke menu utama. *Script* untuk antarmuka *gameover* dapat dilihat pada kode sumber 4.30.

```
void Update ()
{
    if(Input.GetButtonDown("Fire1"))
    {
        int level =
        PlayerPrefs.GetInt ("atLevel");
        SceneManager.LoadScene
        (level);
    }

    if(Input.GetButtonDown("Fire2"))
```

```

    {
        SceneManager.LoadScene ("Main
Menu");
    }
}

```

Kode Sumber 4.30 *Script Antarmuka Gameover*

Lalu apabila pengguna menyelesaikan level dengan cara darah yang dihisap mencapai 100, maka akan muncul antarmuka permainan menang (*level complete*) yang terlihat seperti Gambar 4.37.



Gambar 4.37 Tampilan Antarmuka *Level Complete*

Pada antarmuka ini hanya ada 2 perintah yang dapat dijalankan dengan menekan tombol “A” atau “B” pada kontroller, yaitu melanjutkan ke level selanjutnya atau keluar dari level (*quit game*), tombol “A” untuk melanjutkan ke level selanjutnya, sedangkan “B” untuk kembali ke menu utama. *Script* untuk antarmuka *gameover* dapat dilihat pada kode sumber 4.31.

```

void Update ()
{
    if (Input.GetButtonDown ("Fire1"))

```



```

    {
        int level =
PlayerPrefs.GetInt ("atLevel") + 1;
        PlayerPrefs.SetInt
("atLevel",level + 1);
        SceneManager.LoadScene
(level);
    }
    if (Input.GetButtonDown ("Fire2"))
    {
        SceneManager.LoadScene ("Main
Menu");
    }
}

```

Kode Sumber 4.31 Script Antarmuka Level Complete

4.9.4 Implementasi Antarmuka *Minigame*

Pada tahap ini dijelaskan tentang antarmuka *minigame*. Antarmuka ini akan muncul ketika pengguna menekan tombol “A” pada kontroller dan berada di area tempat menghisap darah objek manusia dan memulai permainan *minigame*. Tampilan antarmuka *minigame* akan terlihat seperti Gambar 4.38.



Gambar 4.38 Tampilan Antarmuka *Minigame*

Tampilan antarmuka *minigame* hanya terdiri dari 4 ikon arah. Ikon arah tersebut berubah warna menjadi hijau apabila ditekan sesuai dengan tombol arah tersebut dan berubah menjadi merah gelap apabila menekan arah yang salah. *Script* untuk menampilkan arah pada *minigame* terlihat pada Kode Sumber 4.31.

```
public IEnumerator MiniGameStart()
{
    Debug.Log("Minigame start");
    int x = Random.Range(0, 9);
    float time = 1f;
    yield return new WaitForSeconds(time);
    while(miniGameStart)
    {
        if (miniGameStart)
        {
            int y = Random.Range(0, 4);
            Debug.Log(y);
            if (y == 4)
            {
                y = Random.Range(0, 3);
            }
            button[y].SetActive(true);
            yield return new
WaitForSeconds(time);
            button[y].SetActive(false);
            button[y].GetComponent<RhythmGameButton>()
).enabled = true;
            button[y].GetComponent<SpriteRenderer>().
color = initialColor;
            yield return new
WaitForSeconds(time);
        }
        else yield return 0;
    }
}
```

Kode Sumber 4.32 Script Menampilkan Arah pada Minigame

BAB V

PENGUJIAN DAN EVALUASI

Bab ini berisi bahasan mengenai uji coba dan evaluasi permainan realitas virtual kehidupan hewan nyamuk dari sudut pandang orang pertama menggunakan Google Cardboard. Uji coba dilakukan menggunakan metode kotak hitam berdasarkan skenario yang telah ditentukan. Uji coba dilakukan terhadap hasil implementasi perangkat lunak yang dijelaskan pada Bab IV.

5.1 Lingkungan Uji Coba

Proses uji coba dilakukan pada ponsel pintar (*smartphone*) Asus Padfone S. Pada uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan lingkungan perangkat lunak. Penjelasan spesifikasi perangkat keras dan perangkat lunak pada lingkungan uji coba dapat dilihat pada Tabel 5.1.

Tabel 5.1 Lingkungan Perangkat Uji Coba

Perangkat Keras	Prosesor : - Qualcomm MSM8974AB Snapdragon 801 @2.3 GHz Quad-Core Memori : - 2048 MB RAM
Perangkat Lunak	Sistem Operasi : - Android Versi 5.0

5.2 Pengujian Fungsionalitas

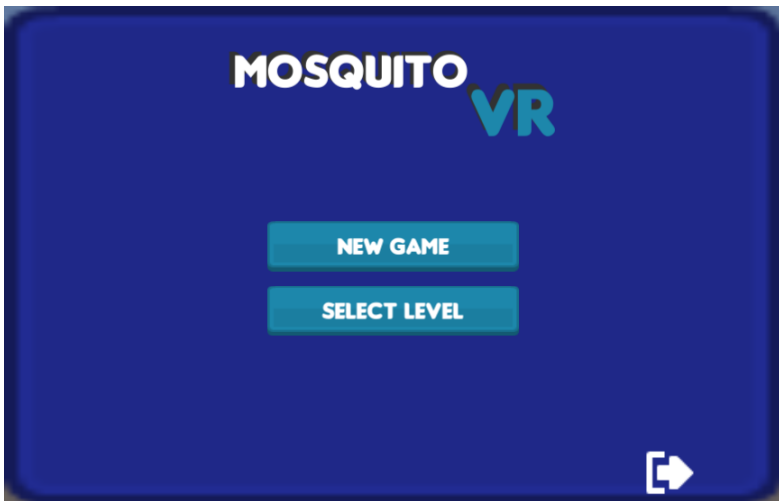
Pengujian fungsionalitas dilakukan untuk mengetahui kesesuaian keluaran dari tiap tahap atau langkah penggunaan fitur terhadap skenario yang telah dipersiapkan. Pengujian akan dilakukan oleh beberapa pengguna. Berikut ini penjabaran skenario dan hasil uji coba fungsionalitas yang dilakukan terhadap permainan realitas virtual yang dibangun.

5.2.1 Skenario Pengujian Memilih Menu Navigasi

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah tombol navigasi dalam menu dapat berfungsi dengan baik. Pada pengujian ini, pengguna cukup menekan tombol yang ada di dalam menu pada layar sentuh untuk mengakses ke menu lainnya atau mengakses level yang ada di game. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.2. dan gambar pengujian yang terlihat pada Gambar 5.1 dan Gambar 5.2.

Tabel 5.2 Pengujian Memilih Menu Navigasi

ID	UJ-UC-001
Nama	Memilih Menu Navigasi
Tujuan uji coba	Mengetahui apakah pengguna dapat memilih menu navigasi saat permainan dijalankan pertama kali
Kondisi awal	Pengguna sudah masuk ke permainan
Masukan	Pengguna menekan tombol yang ada dalam permainan
Keluaran yang diharapkan	Pengguna memasuki menu lain atau memulai memasuki level
Hasil Uji Coba	Berhasil
Kondisi Akhir	Sistem me-load level yang dipilih pengguna



Gambar 5.1 Uji Coba saat Menekan Tombol Navigasi



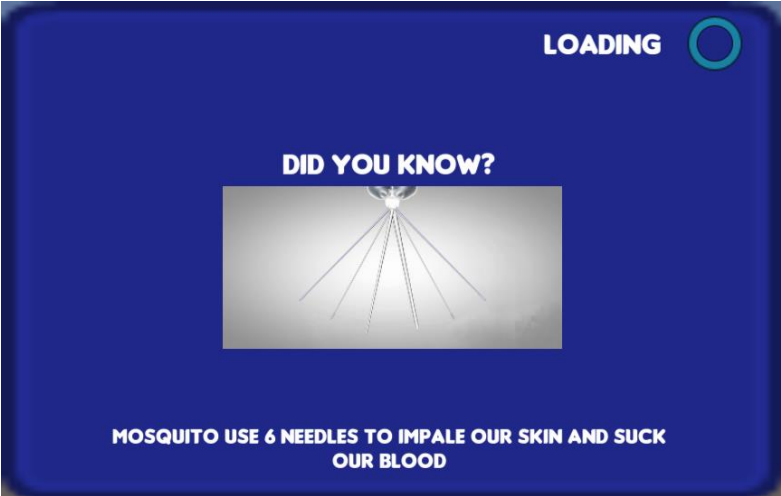
Gambar 5.2 Uji Coba saat Selesai Menekan Tombol Navigasi

5.2.2 Skenario Pengujian Melihat Info Fakta (*Fun Fact*) dari Hewan Nyamuk

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah info fakta tentang nyamuk dapat dimunculkan oleh system dan dilihat oleh pengguna ketika system me-load level yang dipilih pengguna. Pada pengujian ini, pengguna cukup melihat info fakta pada layar ketika permainan dalam antarmuka layar tunggu. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.3. dan gambar pengujian yang terlihat pada gambar 5.3.

Tabel 5.3 Pengujian Melihat Info Fakta (*Fun Fact*) dari Hewan Nyamuk

ID	UJ-UC-002
Nama	Melihat Info Fakta (<i>Fun Fact</i>) dari Hewan Nyamuk
Tujuan uji coba	Mengetahui apakah pengguna dapat melihat informasi tentang fakta dari hewan nyamuk saat permainan me-load level permainan
Kondisi awal	Pengguna sudah memilih level yang dimainkan dan masuk ke dalam antarmuka layar tunggu
Masukan	-
Keluaran yang diharapkan	Pengguna melihat info fakta tentang nyamuk
Hasil Uji Coba	Berhasil
Kondisi Akhir	Info fakta diaktifkan oleh sistem dan terlihat oleh pengguna



Gambar 5.3 Uji Coba saat Melihat Info Fakta Tentang Nyamuk

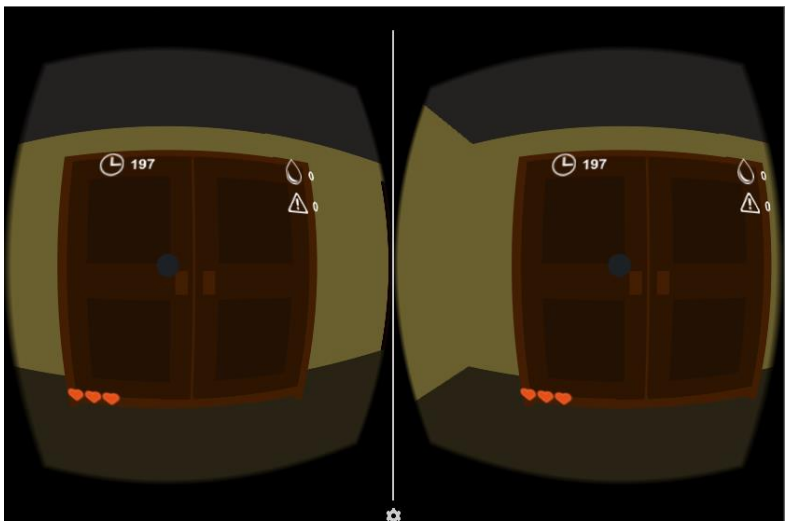
5.2.3 Skenario Pengujian Bergerak atau Terbang Mengelilingi Ruangan

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah tombol untuk bergerak atau terbang dapat berjalan dengan baik dan pergerakan karakter dapat bergerak sesuai keinginan pengguna. Pada pengujian ini, pengguna cukup menekan stik analog kearah yang diinginkan untuk bergerak atau terbang di dalam level permainan. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.4. dan gambar pengujian yang terlihat pada gambar 5.4 dan gambar 5.5.

Tabel 5.4 Pengujian Bergerak atau Terbang Mengelilingi Ruangan

ID	UJ-UC-003
Nama	Bergerak atau Terbang Mengelilingi Ruangan

Tujuan uji coba	Mengetahui apakah tombol bergerak dapat berjalan dengan baik dan pengguna dapat bergerak dengan bebas mengelilingi ruangan dalam permainan
Kondisi awal	Pengguna sudah memasuki level yang diinginkan
Masukan	Pengguna menekan stik analog kea rah yang diinginkan untuk bergerak atau terbang
Keluaran yang diharapkan	Pengguna terbang mengelilingi ruangan pada level yang dimainkan
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pengguna selesai terbang mengelilingi ruangan pada level yang dimainkan



Gambar 5.4 Uji Coba saat Bergerak atau Terbang Mengelilingi Ruangan



Gambar 5.5 Uji Coba saat Selesai Bergerak atau Terbang Mengelilingi Ruangan

5.2.4 Skenario Pengujian Menghisap Darah Musuh atau Target

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah karakter yang dimainkan pengguna dapat menghisap darah musuh atau target dan tombol untuk memulai menghisap darah dapat berjalan dengan baik. Pada pengujian ini, pengguna cukup mendekati ke area menghisap darah target dan menekan tombol untuk memulai menghisap darah di dalam level permainan. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.5 dan gambar pengujian yang terlihat pada gambar 5.6 dan gambar 5.7.

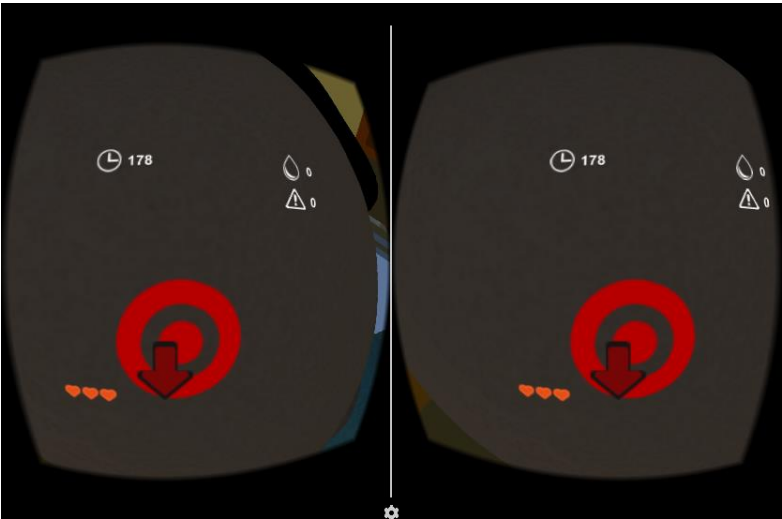
Tabel 5.5 Pengujian Menghisap Darah Musuh atau Target

ID	UJ-UC-004
Nama	Menghisap Darah Musuh atau Target
Tujuan uji coba	Mengetahui apakah Pengguna dapat menghisap

	darah target ketika pengguna berada di daerah tempat hisap darah dari target
Kondisi awal	Pengguna sudah memasuki level yang diinginkan
Masukan	pengguna mendekati ke area menghisap darah target dan menekan tombol untuk memulai menghisap darah
Keluaran yang diharapkan	Pengguna memulai permainan <i>minigame</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pengguna memulai permainan <i>minigame</i>



Gambar 5.6 Uji Coba saat Menghisap Darah Musuh atau Target



Gambar 5.7 Uji Coba saat Selesai Menghisap Darah Musuh atau Target

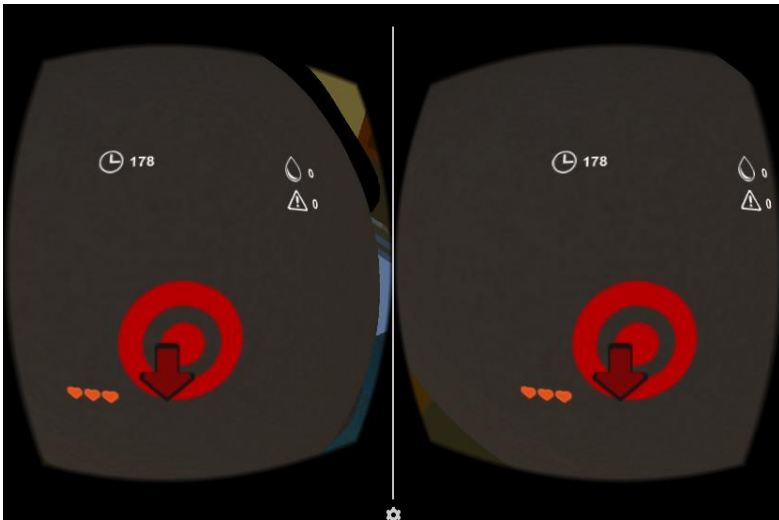
5.2.5 Skenario Pengujian Menyelesaikan *Minigame*

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah tombol dalam *minigame* dapat bekerja dengan baik dan *minigame* dapat diselesaikan oleh pengguna. Pada pengujian ini, pengguna cukup menekan tombol ketika pengguna sudah berada di area menghisap darah target untuk memulai menghisap darah di dalam level permainan. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.6 dan gambar pengujian yang terlihat pada gambar 5.8 dan gambar 5.9.

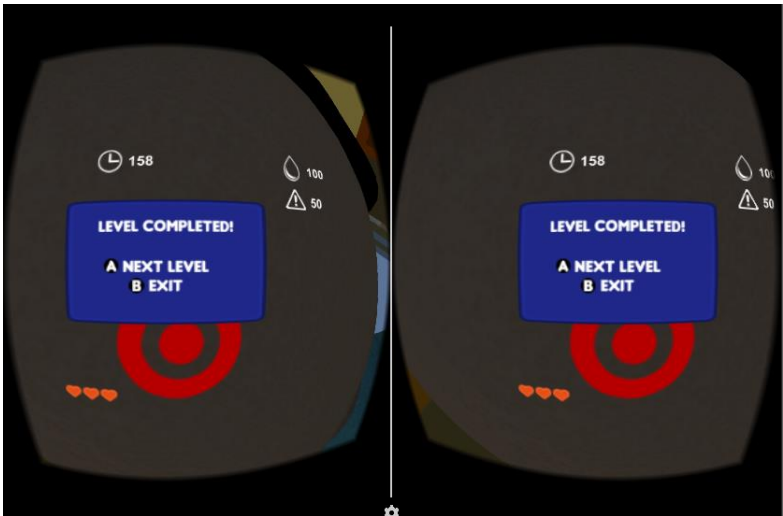
Tabel 5.6 Pengujian Menyelesaikan *Minigame*

ID	UJ-UC-005
Nama	Menyelesaikan <i>Minigame</i>
Tujuan uji coba	Mengetahui apakah Pengguna dapat menjalankan dan menyelesaikan minigame ketika pengguna

	memulai menghisap darah dari target
Kondisi awal	Pengguna sudah memasuki level yang diinginkan dan karakter pengguna sudah berada di area menghisap darah target
Masukan	pengguna menekan tombol untuk memulai menghisap darah
Keluaran yang diharapkan	Pengguna memulai dan permainan <i>minigame</i>
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pengguna menyelesaikan permainan <i>minigame</i> dan permainan selesai



Gambar 5.8 Uji Coba saat Menyelesaikan *Minigame*



Gambar 5.9 Uji Coba saat Selesai Menyelesaikan *Minigame*

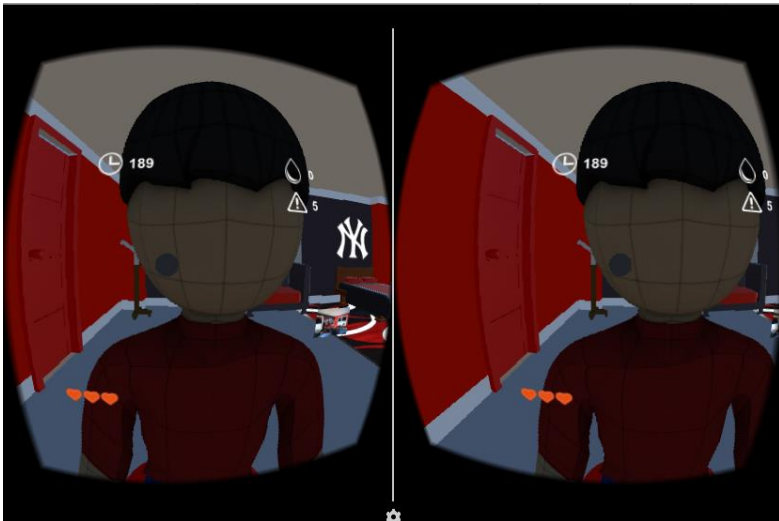
5.2.6 Skenario Pengujian Menghindari Pukulan Musuh atau Target

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah animasi musuh dapat berjalan dengan baik ketika pengguna mengganggu musuh dan pengguna dapat menghindari dari pukulannya. Pada pengujian ini, pengguna terbang ke arah pandangan musuh lalu menghindari ketika musuh berusaha memukul pengguna di dalam level permainan. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.7 dan gambar pengujian yang terlihat pada gambar 5.10 dan gambar 5.11.

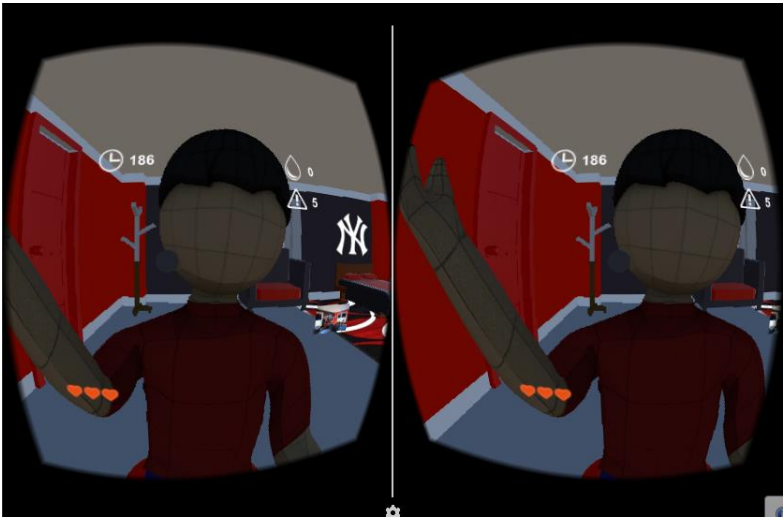
Tabel 5.7 Pengujian Menghindari Pukulan Musuh atau Target

ID	UJ-UC-006
Nama	Menghindari Pukulan Musuh atau Target
Tujuan uji coba	Mengetahui apakah pengguna dapat menghindari

	serangan atau pukulan dari target ketika target merasa terganggu dengan keberadaan pengguna
Kondisi awal	Pengguna sudah memasuki level yang diinginkan
Masukan	Pengguna terbang kearah pandangan musuh dan mengganggu musuh
Keluaran yang diharapkan	Musuh berusaha memukul karakter pengguna dan pengguna berhasil menghindari pukulan musuh
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pengguna berhasil menghindari pukulan musuh dan karakter pengguna tidak berkurang nyawanya



Gambar 5.10 Uji Coba saat Menghindari Pukulan Musuh atau Target



Gambar 5.11 Uji Coba saat Selesai Menghindari Pukulan Musuh atau Target

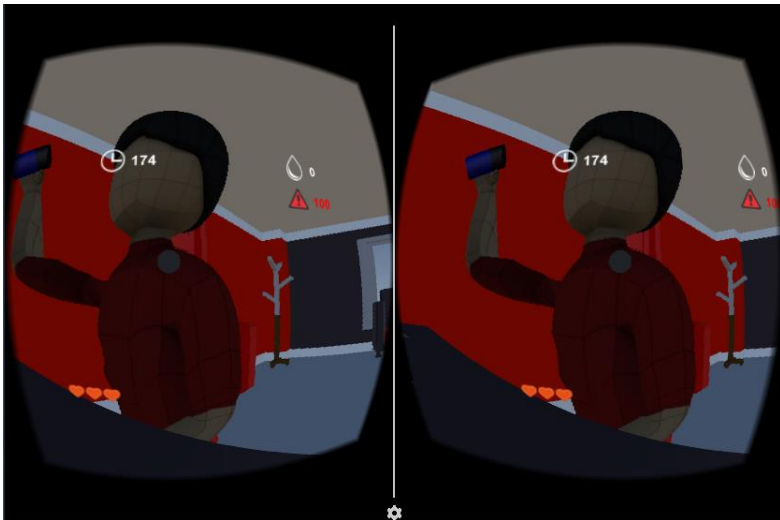
5.2.7 Skenario Pengujian Bersembunyi dari Racun

Skenario pengujian pada tahap ini adalah pengujian yang digunakan untuk mengecek fungsionalitas apakah racun dapat berjalan dengan baik ketika pengguna mengganggu musuh dan tempat untuk bersembunyi dapat melindungi karakter pengguna. Pada pengujian ini, pengguna terbang ke arah pandangan musuh lalu berusaha bersembunyi ke area bersembunyi dari racun. Untuk penjelasan lebih lanjut, skenario akan ditunjukkan pada Tabel 5.7 dan gambar pengujian yang terlihat pada gambar 5.10 dan gambar 5.11.

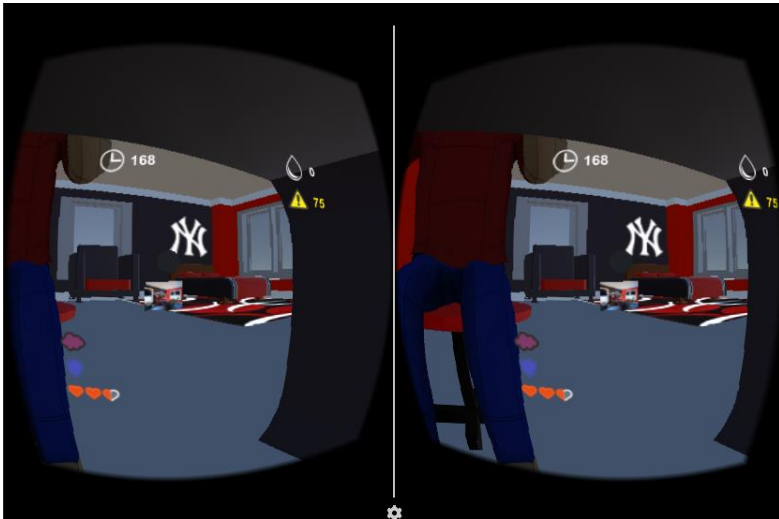
Tabel 5.8 Pengujian Bersembunyi dari Racun

ID	UJ-UC-007
Nama	Bersembunyi dari Racun
Tujuan uji coba	Mengeahui apakah pengguna dapat bersembunyi di tempat yang aman ketika target menggunakan obat nyamuk untuk meyebarakan racun

Kondisi awal	Pengguna sudah memasuki level yang diinginkan
Masukan	Penggun aterbang kearah pandangan musuh dan mengganggu musuh
Keluaran yang diharapkan	Musuh menyembrotkan racun dan racun menyebar dan pengguna berhasil bersembunyi dari terkena racun
Hasil Uji Coba	Berhasil
Kondisi Akhir	Pengguna berhasil bersembunyi di area bersembunyi dari racun



Gambar 5.12 Uji Coba saat Bersembunyi dari Racun



Gambar 5.13 Uji Coba saat Selesai Bersembunyi dari Racun

5.3 Pengujian Aplikasi Terhadap Pengguna

Subbab ini membahas mengenai pengujian pada aplikasi yang tidak hanya menguji fungsionalitas yang ada di dalam permainan tapi juga penggunaan secara langsung oleh orang-orang terkait. Pengujian termasuk ke dalam pengujian subjektif yang bertujuan untuk mengukur kesuksesan perancangan aplikasi menurut garis pandang pengguna. Hal ini bisa didapatkan dengan meminta penilaian serta kritik dan saran melalui kuesioner terhadap beberapa aspek yang tersedia di aplikasi.

5.3.1 Skenario Uji Coba oleh Pengguna

Penguji diminta untuk mencoba seluruh fungsionalitas yang tersedia di permainan ini saat melakukan pengujian permainan. Sebelum permainan diuji oleh pengguna, ada beberapa informasi seputar permainan, kegunaan, dan fitur yang tersedia. Lalu penguji diarahkan untuk langsung mencoba

permainan dengan spesifikasi lingkungan permainan yang sama seperti diterangkan pada lingkungan uji coba.

Jumlah pengguna yang akan menguji permainan sebanyak 5 (lima) orang. Tiap orang akan mendapatkan kuesioner sebagai media untuk memberikan penilaian dan tanggapan serta kritik dan saran pada bagian akhir kuesioner untuk perkembangan aplikasi ke depannya.

5.3.2 Daftar Penguji Permainan

Terdapat lima orang penguji seperti yang telah dijelaskan pada bab sebelumnya. Daftar nama penguji permainan dapat dilihat pada Tabel 5.9.

Tabel 5.9 Daftar Penguji Permainan

Nomor	Nama	Pekerjaan
1	Purbo Panambang	Mahasiswa
2	Saddhana Arta Daniswara	Mahasiswa
3	Ishardan	Mahasiswa
4	Cayza Angka Maulana	Mahasiswa
5	Wildan Lutfi S.F.S	Mahasiswa

5.3.3 Hasil Uji Coba Pengguna

Uji coba yang dilakukan oleh pengguna akan mengukur beberapa aspek pada permainan. Aspek yang diukur adalah kenyamanan antarmuka, lingkungan realitas virtual, dan penggunaan *google cardboard* dan controller, kemudahan dalam permainan, suasana permainan, dan performa permainan dan kontrol. Sistem penilaian berdasarkan pada skala perhitungan satu sampai enam. Skala satu menunjukkan nilai terendah (sangat tidak setuju) dan skala enam menunjukkan nilai tertinggi (sangat setuju). Tabel 5.10 menunjukkan skala dan keterangan dari skala tersebut. Penilaian akhir dilakukan dengan memperlihatkan

seberapa banyak penguji memilih salah satu skala dari tiap enam skala yang ada.

Tabel 5.10 Skala Penilaian Uji Coba

Skala	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Kurang Setuju
4	Cukup Setuju
5	Setuju
6	Sangat Setuju

5.3.3.1 Hasil Penilaian Kenyamanan Antarmuka dan Lingkungan Realitas Virtual

Penilaian ini berfokuskan kenyamanan penguji dalam antarmuka dan lingkungan yang ada di dalam permainan. Hasil penilaian penguji ditunjukkan pada Tabel 5.11.

Tabel 5.11 Penilaian Kenyamanan Antarmuka dan Lingkungan Realitas Virtual

No.	Indikator	Penilaian					
		1	2	3	4	5	6
1	Kenyamanan antarmuka	0	0	0	2	2	1
2	Kenyamanan lingkungan	0	0	0	1	3	1
3	Kenyamaan <i>cardboard</i> dan kottroller	0	0	1	0	3	1

Hasil pengujian menunjukkan bahwa semua merasa nyaman dengan antarmuka dan lingkungan yang ada didalam permainan realitas virtual. Tetapi untuk kenyamanan penggunaan

google cardboard dan kontroller terdapat 1 orang yang merasa tidak nyaman dan 4 orang lainnya merasa nyaman.

5.3.3.2 Hasil Penilaian Kemudahan Permainan

Penilaian ini berfokuskan kemudahan dalam memahami dan menyelesaikan tugas yang diberikan dalam permainan. Hasil penilaian penguji ditunjukkan pada Tabel 5.12.

Tabel 5.12 Penilaian Kemudahan Permainan

No.	Indikator	Penilaian					
		1	2	3	4	5	6
1	Kemudahan memahami permainan	0	0	0	3	1	1
2	Kemudahan menjalankan tugas permainan	0	0	0	2	2	1

Hasil pengujian menunjukkan bahwa semua merasa mudah dalam memahami maupun menjalankan tugas dalam permainan.

5.3.3.3 Hasil Penilaian Suasana Permainan

Penilaian ini berfokuskan suasana yang menyerupai lingkungan dunia nyata dalam permainan. dan karakter dalam permainan. Hasil penilaian penguji ditunjukkan pada Tabel 5.13.

Tabel 5.13 Penilaian Suasana Permainan

No.	Indikator	Penilaian					
		1	2	3	4	5	6
1	Suasana lingkungan permainan	0	0	0	1	2	2
2	Suasana karakter permainan	0	0	1	1	2	0

Hasil pengujian menunjukkan bahwa semua merasa suasana lingkungan dalam permainan realitas virtual yang dibangun serasa seperti lingkungan di dunia nyata, tetapi untuk karakter terdapat 1 orang yang merasa belum seperti karakter di dunia nyata.

5.3.3.4 Hasil Penilaian Performa Pemain

Penilaian ini berfokuskan performa dan kontrol dalam permainan. Hasil penilaian pengujian ditunjukkan pada Tabel 5.13.

Tabel 5.14 Penilaian Performa Permainan

No.	Indikator	Penilaian					
		1	2	3	4	5	6
1	Performa permainan	0	0	0	2	1	2
2	Performa kontrol permainan	0	0	2	2	1	0

Hasil pengujian menunjukkan bahwa semua merasa performa permainan cukup baik tetapi untuk kontrol permainan terdapat 2 orang yang merasa kontrol kurang baik.

5.4 Evaluasi

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian fungsionalitas dan pengujian non-fungsionalitas yang telah dilakukan sebelumnya.

5.4.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan pada subbab 5.2. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.15. Dari data

yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.15 Rekapitulasi Hasil Uji Coba Fungsionalitas

ID	Indikator	Hasil
UJ-UC-001	Uji Coba Memilih Menu Navigasi	Berhasil
UJ-UC-002	Uji Coba Melihat Info Fakta (<i>Fun Fact</i>) dari Hewan Nyamuk	Berhasil
UJ-UC-003	Uji Coba Bergerak atau Terbang Mengelilingi Ruangan	Berhasil
UJ-UC-004	Uji Coba Menghisap Darah Musuh atau Target	Berhasil
UJ-UC-005	Uji Coba Menyelesaikan Minigame	Berhasil
UJ-UC-006	Uji Coba Menghindari Pukulan Musuh atau Target	Berhasil
UJ-UC-007	Uji Coba Bersembunyi dari Racun	Berhasil

5.4.2 Evaluasi Pengujian Permainan

Pengujian dilakukan oleh 5 orang mahasiswa Teknik Informatika ITS dengan menggunakan system mayoritas (tanpa rata-rata). Jawaban kuesioner dari tiap pengguna dapat dilihat pada lampiran. Rangkuman hasil uji coba pengguna dapat dilihat pada tabel 5.16.

Tabel 5.16 Rangkuman Uji Coba Pengguna

Penilaian	Indikator	Penilaian Mayoritas
Kenyamanan Antarmuka dan Lingkungan	Kenyamanan antarmuka	Cukup Puas (3 dari 5 responden)
	Kenyamanan lingkungan	Puas (3 dari 5 responden)
	Kenyamaan <i>cardboard</i> dan kontroller	Puas (3 dari 5 responden)

Penilaian	Indikator	Penilaian Mayorias
Total Penilaian		Puas
Kemudahan Permainan	Kemudahan memahami permainan	Cukup Puas (3 dari 5 responden)
	Kemudahan menjalankan tugas permainan	Puas (4 dari 5 responden)
Total Penilaian		Puas
Suasana Permainan	Suasana lingkungan permainan	Puas (4 dari 5 responden)
	Suasana karakter permainan	Puas (3 dari 5 responden)
Total Penilaian		Puas
Performa Permainan	Performa permainan	Puas (3 dari 5 responden)
	Performa kontrol permainan	Cukup Puas (4 dari 5 responden)
Total Penilaian		Cukup Puas

Hasil rangkuman diatas menunjukkan dari penilaian yang ada memiliki tingkat kepuasan cukup puas dan puas. Kesimpulan dari evaluasi ini adalah pengguna merasa puas terhadap suasana dan kenyamanan dalam permainan, dan merasa terdapat fitur yang perlu diperbaiki untuk kemudahan dan performa permainan. Untuk rangkuman kritik dan saran pengguna dapat dilihat pada Tabel 5.17.

Tabel 5.17 Rangkuman Uji Coba Pengguna

Nama	Kritik dan Saran
Purbo Panambang	Kontroller untuk maju dan mundur sedikit kurang nyaman

Saddhana Arta Daniswara	Perlu diberi tutorial, dan ketika disemprotkan obat nyamuk diperlukan intruksi untuk bersembunyi
Ishardan	Maju mundur dibuat otomatis
Cayza Angka Maulana	Asset-asset diperindah
Wildan Lutfi S.F.S	Kontrol untuk mengarah ke bawah kurang menurun

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan permainan dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Orang Pertama menggunakan Google Cardboard berhasil dibuat menggunakan aplikasi Unity3D dengan menggunakan Bahasa C# dengan objek yang dibuat dengan aplikasi Blender.
2. Dari kebutuhan fungsional pada Tabel 3.2, permainan simulasi kehidupan hewan nyamuk berhasil menjalankan semua kebutuhan yang dibuat. Berdasarkan Tabel 5.15, dapat disimpulkan semua fungsionalitas aplikasi berfungsi dengan baik.
3. Berdasarkan pengujian permainan, sebagian penguji merasa nyaman dan mudah dalam memainkan permainan dan merasa suasana dalam permainan seperti dalam dunia nyata.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Permainan yang dibangun perlu ditambahkan tutorial untuk memudahkan pengguna saat pertama kali memainkan permainan.
2. Asset-asset permainan diperindah agar lebih terlihat nyata.
3. Kontrol permainan perlu diperbaiki lagi agar lebih nyaman digunakan.

DAFTAR PUSTAKA

- [1] “Virtual reality,” *Wikipedia*. 03-Jun-2017.
- [2] “VR Game List,” *VR Games For*. [Daring]. Tersedia pada: <http://vrgamesfor.com/list/>. [Diakses: 05-Jun-2017].
- [3] “Colonizing Virtual Reality.” [Daring]. Tersedia pada: http://cultronix.eserver.org/chesher/?utm_source=friendfeedlikes&utm_medium=twitter. [Diakses: 05-Jun-2017].
- [4] “Google Cardboard Definition from PC Magazine Encyclopedia.” [Daring]. Tersedia pada: <http://www.pcmag.com/encyclopedia/term/67932/google-cardboard>. [Diakses: 05-Jun-2017].
- [5] “Introduction to Unity3D.” [Daring]. Tersedia pada: <https://code.tutsplus.com/tutorials/introduction-to-unity3d--mobile-10752>. [Diakses: 05-Jun-2017].
- [6] A. Brito, *Blender 3D : Architecture, Buildings, and Scenery*. Birmingham: PACKT publishing, 2008.
- [7] “Mister Mosquito,” *Wikipedia*. 09-Mei-2017.
- [8] “The Top 7... games that are cheaper than therapy: Page 3 | GamesRadar+.” [Daring]. Tersedia pada: <http://www.gamesradar.com/the-top-7-games-that-are-cheaper-than-therapy/3/>. [Diakses: 05-Jun-2017].
- [9] “Three-dimensional space,” *Wikipedia*. 01-Mei-2017.
- [10] “What is First Person Shooter (FPS)? - Definition from Techopedia.” [Daring]. Tersedia pada: <https://www.techopedia.com/definition/241/first-person-shooter-fps>. [Diakses: 05-Jun-2017].
- [11] L. Webster, “Visual Studio IDE, Code Editor, Team Services, & Mobile Center,” *Visual Studio*, 28-Apr-2017. [Daring]. Tersedia pada: <https://www.visualstudio.com/>. [Diakses: 05-Jun-2017].
- [12] “Nyamuk,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 04-Jun-2017.

(Halaman Ini Sengaja dikosongkan)

LAMPIRAN HASIL KUESIONER

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Pertama menggunakan Google Cardboard

Nama : Saddhana Arta Daniswara
Umur : 19
Pekerjaan : Mahasiswa

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah anda pernah menggunakan Google Cardboard sebelumnya?	(a.) Ya	b. Tidak
2	Apakah anda pernah memainkan game bergenre <i>first person shooter</i> ?	(a.) Ya	b. Tidak
3	Apakah anda pernah memainkan game tentang aktifitas dari kehidupan hewan?	(a.) Ya	b. Tidak

Kuesioner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4 : Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Task	Nilai					
		1	2	3	4	5	6
1	Saya merasa nyaman dengan antarmuka/user interface permainan ini				✓		
2	Saya merasa nyaman dengan melihat lingkungan realitas virtual pada permainan ini						✓
3	Saya merasa nyaman memainkan permainan ini menggunakan cardboard dan Bluetooth controller secara bersamaan					✓	
4	Saya merasa mudah memahami permainan ini				✓		
5	Saya merasa mudah dalam menjalankan task/aktivitas yang diberikan				✓		
6	Saya merasa lingkungan permainan ini mirip dengan lingkungan dunia nyata						✓
7	Saya merasa seperti menjadi karakter utama (hewan) dalam permainan ini					✓	
8	Saya merasa performa permainan baik						✓
9	Saya merasa kontrol permainan baik			✓			

Usability Testing:

Berikut ini isi waktu pencapaian pada aktifitas yang dijalankan

No	Task	Kondisi Selesai	Waktu Pencapaian
1	Memilih Menu Navigasi	Pemain memasuki level permainan	2
2	Melihat info fakta (<i>fun fact</i>) dari hewan nyamuk	1. Sistem mempersiapkan level permainan 2. Pemain melihat info fakta	2
3	bergerak (terbang) mengelilingi ruangan		25

Gambar 8.1 Lembar Kuesioner Saddhana Arta Daniswara (1)

No	Kategori	Detail	Nilai
4	Menghisap darah musuh/target	1. Pemain berada di posisi tempat menghisap darah 2. Pemain memulai mini game menghisap darah	20
5	menyelesaikan mini game	Pemain menyelesaikan mini game menghisap darah	25
6	Menghindari pukulan musuh/target	1. Pemain berada di posisi pandangan musuh beberapa selang waktu 2. Pemain menghindari pukulan musuh	17
7	bersembunyi dari racun	1. Musuh menyebarkan racun 2. Pemain bersembunyi di tempat bersembunyi 3. Nyawa pemain berhenti berkurang dan racun berhenti menyebar	19



Kritik dan Saran untuk pengembangan selanjutnya

Kontrolnya sudah cukup baik, Namun masih kurang tutorial, dan ketika di sempatkan oleh nyawa tidak ada instruksi tempat bersembunyi

Surabaya, 3 Juni 2017

(Saddhona Arta D.)

Gambar 8.2 Lembar Kuesioner Saddhana Arta Daniswara (2)

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Pertama menggunakan Google Cardboard

Nama : ISHARDAN
Umur : 5113100181
Pekerjaan : Mahasiswa

Silahkan lingkari (O) padapilihan yang sesuai

1. Apakah anda pernah menggunakan Google Cardboard sebelumnya?	<input checked="" type="radio"/> Ya	b. Tidak
2. Apakah anda pernah memainkan game bergenre <i>first person shooter</i> ?	<input checked="" type="radio"/> Ya	b. Tidak
3. Apakah anda pernah memainkan game tentang aktifitas dari kehidupan hewan?	<input checked="" type="radio"/> Ya	b. Tidak

Kuisloner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1:Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4 : Cukup Setuju, 5: Setuju, 6:Sangat Setuju

No	Task	Nilai					
		1	2	3	4	5	6
1	Saya merasa nyaman dengan antarmuka/user interface permainan ini					✓	
2	Saya merasa nyaman dengan melihat lingkungan realitas virtual pada permainan ini				✓		
3	Saya merasa nyaman memainkan permainan ini menggunakan cardboard dan Bluetooth controller secara bersamaan			✓			
4	Saya merasa mudah memahami permainan ini				✓		
5	Saya merasa mudah dalam menjalankan task/aktivitas yang diberikan				✓		
6	Saya merasa lingkungan permainan ini mirip dengan lingkungan dunia nyata				✓		
7	Saya merasa seperti menjadi karakter utama (hewan) dalam permainan ini			✓			
8	Saya merasa performa permainan baik					✓	
9	Saya merasa kontrol permainan baik			✓			

Usability Testing:

Berikut ini isi waktu pencapaian pada aktifitas yang dijalankan

No	Task	Kondisi Selesai	Waktu Pencapaian
1	Memilih Menu Navigasi	Pemain memasuki level permainan	
2	Melihat Info fakta (<i>fun fact</i>) dari hewan nyamuk	1. Sistem mempersiapkan level permainan 2. Pemain melihat info fakta	2 detik
3	bergerak (terbang) mengelilingi ruangan		

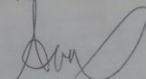
Gambar 8.3 Lembar Kuesioner Ishardan (1)

ITS		ITS	
4	Menghisap darah musuh/target	1. Pemain berada di posisi tempat menghisap darah 2. Pemain memulai mini game menghisap darah	1 detik
5	menyelesaikan mini game	Pemain menyelesaikan mini game menghisap darah	3 detik
6	Menghindari pukulan musuh/target	1. Pemain berada di posisi pandangan musuh beberapa selang waktu 2. Pemain menghindari pukulan musuh	—
7	bersembunyi dari racun	1. Musuh menyebarkan racun 2. Pemain bersembunyi di tempat bersembunyi 3. Nyawa pemain berhenti berkurang dan racun berhenti menyebar	—



Kritik dan Saran untuk pengembangan selanjutnya

Mau gymkit secara otomatis
Kekurangan sudah baik

Surabaya, 31 Mei 2017

()

Gambar 8.4 Lembar Kuesioner Ishardan (2)

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Pertama menggunakan Google Cardboard

Nama : Wildan Lutfi S.F.S
Umur : 20
Pekerjaan : Mahasiswa

Silahkan lingkari (O) pada pilihan yang sesuai

1. Apakah anda pernah menggunakan Google Cardboard sebelumnya?	a. Ya	b. Tidak
2. Apakah anda pernah memainkan game bergenre <i>first person shooter</i> ?	a. Ya	b. Tidak
3. Apakah anda pernah memainkan game tentang aktifitas dari kehidupan hewan?	a. Ya	b. Tidak

Kuisloner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1:Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4 : Cukup Setuju, 5: Setuju, 6:Sangat Setuju

No	Task	Nilai					
		1	2	3	4	5	6
1	Saya merasa nyaman dengan antarmuka/user interface permainan ini				✓		
2	Saya merasa nyaman dengan melihat lingkungan realitas virtual pada permainan ini					✓	
3	Saya merasa nyaman memainkan permainan ini menggunakan cardboard dan Bluetooth controller secara bersamaan						✓
4	Saya merasa mudah memahami permainan ini						✓
5	Saya merasa mudah dalam menjalankan task/aktivitas yang diberikan					✓	
6	Saya merasa lingkungan permainan ini mirip dengan lingkungan dunia nyata					✓	
7	Saya merasa seperti menjadi karakter utama (hewan) dalam permainan ini				✓		
8	Saya merasa performa permainan baik				✓		
9	Saya merasa kontrol permainan baik				✓		

Usability Testing:

Berikut ini isi waktu pencapaian pada aktifitas yang dijalankan

No	Task	Kondisi Selesai	Waktu Pencapaian
1	Memilih Menu Navigasi	Pemain memasuki level permainan	2
2	Melihat info fakta (<i>fun fact</i>) dari hewan nyamuk	1. Sistem mempersiapkan level permainan 2. Pemain melihat info fakta	3
3	bergerak (terbang) mengelilingi ruangan		65

Gambar 8.5 Lembar Kuesioner Wildan Lutfi S.F.S (1)

ITS
Institut Teknologi Sepuluh Nopember

Logo of ITS

4	Menghisap darah musuh/target	1. Pemain berada di posisi tempat menghisap darah 2. Pemain memulai mini game menghisap darah	65
5	menyelesaikan mini game	Pemain menyelesaikan mini game menghisap darah	72
6	Menghindari pukulan musuh/target	1. Pemain berada di posisi pandangan musuh beberapa selang waktu 2. Pemain menghindari pukulan musuh	32
7	bersembunyi dari racun	1. Musuh menyebarkan racun 2. Pemain bersembunyi di tempat bersembunyi 3. Nyawa pemain berhenti berkurang dan racun berhenti menyebar	21



Kritik dan Saran untuk pengembangan selanjutnya

Control bagian mengenai kebanahan, kurang

Surabaya, 3 Juni 2017

(Wildan)

Gambar 8.6 Lembar Kuesioner Wildan Lutfi S.F.S (2)

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Pertama menggunakan Google Cardboard

Nama : Purbo Panambang
Umur : 22
Pekerjaan : Mahasiswa

Silahkan lingkari (O) pada pilihan yang sesuai

1. Apakah anda pernah menggunakan Google Cardboard sebelumnya?	(a) Ya	b. Tidak
2. Apakah anda pernah memainkan game bergenre <i>first person shooter</i> ?	(a) Ya	b. Tidak
3. Apakah anda pernah memainkan game tentang aktifitas dari kehidupan hewan?	(a) Ya	b. Tidak

Kuisiner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1:Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4 : Cukup Setuju, 5: Setuju, 6:Sangat Setuju

No	Task	Nilai					
		1	2	3	4	5	6
1	Saya merasa nyaman dengan antarmuka/user interface permainan ini						✓
2	Saya merasa nyaman dengan melihat lingkungan realitas virtual pada permainan ini						✓
3	Saya merasa nyaman memainkan permainan ini menggunakan cardboard dan Bluetooth controller secara bersamaan					✓	
4	Saya merasa mudah memahami permainan ini				✓		
5	Saya merasa mudah dalam menjalankan task/aktivitas yang diberikan					✓	
6	Saya merasa lingkungan permainan ini mirip dengan lingkungan dunia nyata					✓	
7	Saya merasa seperti menjadi karakter utama (hewan) dalam permainan ini				✓		
8	Saya merasa performa permainan baik						✓
9	Saya merasa kontrol permainan baik						✓

Usability Testing:

Berikut ini isi waktu pencapaian pada aktifitas yang dijalankan

No	Task	Kondisi Selesai	Waktu Pencapaian
1	Memilih Menu Navigasi	Pemain memasuki level permainan	1 detik
2	Melihat info fakta (<i>fun fact</i>) dari hewan nyamuk	1. Sistem mempersiapkan level permainan 2. Pemain melihat info fakta	2 detik
3	bergerak (terbang) mengelilingi ruangan		10 detik

Gambar 8.7 Lembar Kuesioner Purbo Panambang (1)

ITS		ITS	
4	Menghisap darah musuh/target	1. Pemain berada di posisi tempat menghisap darah 2. Pemain memulai mini game menghisap darah	6 detik
5	menyelesaikan mini game	Pemain menyelesaikan mini game menghisap darah	5 detik
6	Menghindari pukulan musuh/target	1. Pemain berada di posisi pandangan musuh beberapa selang waktu 2. Pemain menghindari pukulan musuh	-
7	bersembunyi dari racun	1. Musuh menyebarkan racun 2. Pemain bersembunyi di tempat bersembunyi 3. Nyawa pemain berhenti berkurang dan racun berhenti menyebar	-



Kritik dan Saran untuk pengembangan selanjutnya

Sesuai keseluruhan sudah bagus namun controler untuk maju... dan mundur sedikit kurang nyaman

Surabaya, 31 Mei 2017

[Signature]
(Purbo Panambang)

Gambar 8.8 Lembar Kuesioner Purbo Panambang (2)

Permainan Realitas Virtual Kehidupan Hewan Nyamuk dari Sudut Pandang Pertama menggunakan Google Cardboard

Nama : Cayza Angka Maulana
Umur : 21
Pekerjaan : Mahasiswa

Silahkan lingkari (O) pada pilihan yang sesuai

1. Apakah anda pernah menggunakan Google Cardboard sebelumnya?	(a.) Ya	b. Tidak
2. Apakah anda pernah memainkan game bergenre <i>first person shooter</i> ?	(a.) Ya	b. Tidak
3. Apakah anda pernah memainkan game tentang aktifitas dari kehidupan hewan?	a. Ya	(b.) Tidak

Kuisiner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Task	Nilai					
		1	2	3	4	5	6
1	Saya merasa nyaman dengan antarmuka/user interface permainan ini					✓	
2	Saya merasa nyaman dengan melihat lingkungan realitas virtual pada permainan ini					✓	
3	Saya merasa nyaman memainkan permainan ini menggunakan cardboard dan Bluetooth controller secara bersamaan					✓	
4	Saya merasa mudah memahami permainan ini						✓
5	Saya merasa mudah dalam menjalankan task/aktivitas yang diberikan						✓
6	Saya merasa lingkungan permainan ini mirip dengan lingkungan dunia nyata					✓	
7	Saya merasa seperti menjadi karakter utama (hewan) dalam permainan ini					✓	
8	Saya merasa performa permainan baik					✓	
9	Saya merasa kontrol permainan baik					✓	

Usability Testing:

Berikut ini isi waktu pencapaian pada aktifitas yang dijalankan

No	Task	Kondisi Selesai	Waktu Pencapaian
1	Memilih Menu Navigasi	Pemain memasuki level permainan	1 detik
2	Melihat info fakta (<i>fun fact</i>) dari hewan nyamuk	1. Sistem mempersiapkan level permainan 2. Pemain melihat info fakta	2 detik
3	bergerak (terbang) mengelilingi ruangan		15 detik

Gambar 8.9 Lembar Kuesioner Cayza Angka Maulana (1)

ITS Institut Teknologi Sepuluh Nopember		Logo ITS	
4	Menghisap darah musuh/target	1. Pemain berada di posisi tempat menghisap darah 2. Pemain memulai mini game menghisap darah	15 detik.
5	menyelesaikan mini game	Pemain menyelesaikan mini game menghisap darah	2"50"
6	Menghindari pukulan musuh/target	1. Pemain berada di posisi pandangan musuh beberapa selang waktu 2. Pemain menghindari pukulan musuh	20 detik.
7	bersembunyi dari racun	1. Musuh menyebarkan racun 2. Pemain bersembunyi di tempat bersembunyi 3. Nyawa pemain berhenti berkurang dan racun berhenti menyebar	10 detik.

Kritik dan Saran untuk pengembangan selanjutnya

asset-asset dipindahkan

.....

Surabaya, 30 Mei2017

(Cayza Angka M.)

Gambar 8.10 Lembar Kuesioner Cayza Angka Maulana (2)

BIODATA PENULIS



Ken Genesisius Meta Basundara NRP 5113100114, lahir pada 17 Mei 1995 di Sidoarjo, merupakan anak kedua dari dua bersaudara. Penulis pernah menempuh pendidikan formal di SDN Kemiri Sidoarjo SMPN 1 Buduran, dan SMA Antartika Sidoarjo. Saat ini sedang menempuh pendidikan perguruan tinggi di Institut Teknologi Sepuluh Nopember Surabaya di jurusan Teknik Informatika Fakultas Teknologi Informasi angkatan 2013. Memiliki ketertarikan di bidang Teknologi Informasi khususnya pada bidang Interaksi, Grafik, dan Seni. Penulis yang memiliki hobi bermain *game* dan mempelajari hal-hal baru ini merupakan mahasiswa yang cukup aktif dalam berorganisasi. Organisasi yang diikuti yaitu Himpunan Mahasiswa Teknik Computer – Informatika ITS (HTMC), panitia Schematics HMTTC, Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi ITS, Unit Kegiatan Tari dan Karawitan, panitia Generasi Integralistik ITS, dan masih banyak lagi. Penulis dapat dihubungi melalui surat elektronik mkengenesius@gmail.com.